TMS320C6670

Multicore Fixed and Floating-Point System-on-Chip

Data Manual

Literature Number: SPRS689D
March 2012

Release History

Revision	Date	Description/Comments
D	March 2012	Updated PASS PLL section (block diagram, PASS PLL Control Register, and initialization sequence) Updated Switch Fabric Matrix tables with bridge numbers and added Switch Fabric block diagrams Updated the JTAGID register table Restricted Output_Divide of SECCTL to max value of divide by 2 Changed TPTCn to EDMA3TCn and TPCCn to EDMA3CCn throughout the data manual Replaced all INTC with CIC and CPT with Tracer throughout the document Updated main PLL lock time Added DDR3PLL and PASS PLL Reset bits in DDR3PLLCTL1 and PASSPLLCTL1 registers Added the DDR3PLL and PASSPLL Initialization Sequence Added po_vcon_smpserr_intr SmartReflex event Corrected the SPI and DDR3/Hyperbridge Config Memory Map end address
C	October 2011	Added DEVSPEED Register section Removed Parameter Information section from chapter 7 as the content was not relevant Added more description to Boot Sequence section Changed all footnote references from CORECLK to SYSCLK1 Corrected the typo in the address of MACID2 Re-arranged the wording for description of SYSCLK1 Removed example from footnote Updated footnote on AIF jitter value to 4 ps RMS
B	August 2011	Revised the INTC1 Events Input table, description for BWADJ field, and power sequencing timing tables and diagrams Removed all mentions of HHV and the Max parameters for PHY Sync and Radio Sync Pulses Updated the GMacs and GFlops for 1.2 GHz and changed output skew time for the trace from 500 ps to 1ns Added thermal values to the thermal resistance characteristics table, and Power Supply to Peripheral I/O Mapping table Added register and field description table for DDR3PLLCTL1, PASSPLLCTL1, and SerDes status and config registers Corrected RESET electrical timing parameters Updated all PLL block Diagrams – Main PLL, DDR PLL, and PASS PLL Completed all tables in Device Operating Conditions chapter Updated/Added Master and Priv ID tables, added MasterID Settings table Added MMR space
A	April 2011	Updated the power-up sequencing section. $\overline{\text{RESETFULL}}$ must always de-assert after $\overline{\text{POR}}$ Updated the description of VARIANT bit field in JTAGID register Added Setup and Hold times for RP1CLK and RP1CLK signals, and BWADJ field to DDR3PLLCTL and PASSPLLCTL Corrected the size of TETBs for the 4 cores from 16k to 4k Added RSV0A and RSV0B pins to the terminal list table Revised power rail terminology and changed reference parameter in t2c description from t7 to t6 Added a note on Level Interrupts and EOI values for various modules Corrected the address range for I^2C MMRs and corrected extended temp max to 100C from 105C
SPRS689	February 2011	Initial Release

For detailed revision information, see "Revision History" on page A-219.

Contents

List of Figures

List of Tables

1 TMS320C6670 Features

- Four TMS320C66x™ DSP Core Subsystems, Each With
  - 1.0-GHz or 1.2-GHz C66x Fixed/Floating-Point DSP Core
    - 38.4 GMacs/Core for Fixed Point @ 1.2 GHz
    - 19.2 GFlops/Core for Floating Point @ 1.2 GHz
  - Memory
    - 32K Byte L1P Per Core
    - 32K Byte L1D Per Core
    - 1024K Byte Local L2 Per Core
- Multicore Shared Memory Controller (MSMC)
  - 2048KB MSM SRAM Memory Shared by Four DSP Cores
  - Memory Protection Unit for Both MSM SRAM and DDR3_EMIF
- Hardware Coprocessors
  - Three Enhanced Coprocessors for Turbo Decoding
    - Supports WCDMA/HSPA/HSPA+/TD-SCDMA, LTE, and WiMAX
    - Supports up to 548 Mbps for LTE and up to 353 Mbps for WCDMA
    - Low DSP Overhead – HW Interleaver Table Generation and CRC Check
  - One Enhanced Coprocessor for Turbo Encoding
    - Supports up to 500 Mbps for LTE and WCDMA
  - Four Viterbi Decoders
    - Supports More Than 38 Mbps @ 40-bit Block Size
  - Two WCDMA Receive Acceleration Coprocessors
    - Up to 256 Users @ 8 Fingers w/o Measurement
  - WCDMA Transmit Acceleration Coprocessor
    - Up to 256 Users with two Radio Links and Diversity
  - Three Fast Fourier Transform Coprocessors
    - 2048 pt FFT in 4.8 µs
  - Bit Rate Coprocessor
    - WCDMA/HSPA+, TD-SCDMA, LTE, and WiMAX Uplink and Downlink Bit Processing
    - Includes Encoding, Rate Matching/Dematching, Segmentation, Multiplexing, and More
    - Supports Up To 914 Mbps for LTE and 405 Mbps for WCDMA/TD-SCDMA
- Multicore Navigator
  - 8192 Multipurpose Hardware Queues with Queue Manager
  - Packet-Based DMA for Zero-Overhead Transfers
- Network Coprocessor
  - Packet Accelerator Enables Support for
    - Transport Plane IPsec, GTP-U, SCTP, PDCP
    - L2 User Plane PDCP (RoHC, Air Ciphering)
    - 1 Gbps Wire-Speed Throughput at 1.5 M Packets Per Second
  - Security Accelerator Engine Enables Support for
    - IPSec, SRTP, 3GPP and WiMAX Air Interface, and SSL/TLS Security
    - ECB, CBC, CTR, F8, A5/3, CCM, GCM, HMAC, CMAC, GMAC, AES, DES, 3DES, Kasumi, SNOW 3G, SHA-1, SHA-2 (256-bit Hash), MD5
    - Up to 2.8 Gbps Encryption Speed
- Four Rake/Search Accelerators (RSA) for
  - Chip-Rate Processing for WCDMA Rel'99, HSDPA, and HSDPA+
  - Reed-Muller Decoding
- Peripherals
  - Six-Lane SerDes-Based Antenna Interface (AIF2)
    - Operating at up to 6.144 Gbps
    - Compliant with OBSAI RP3 and CPRI Standards for 3G / 4G (WCDMA, LTE TDD, LTE FDD, TD-SCDMA, and WiMAX)
  - Four Lanes of SRIO 2.1
    - 5 GBaud Operation Per Lane
    - Supports Direct I/O, Message Passing
  - Two Lanes PCIe Gen2
    - Supports Up To 5 GBaud Per Lane
  - Hyperlink
    - Supports Connections to Other KeyStone Architecture Devices Providing Resource Scalability
    - Supports up to 50 Gbaud
  - Gigabit Ethernet (GbE) Switch Subsystem
    - Two SGMII Ports
    - IEEE1588 Support
  - 64-Bit DDR3 Interface with Speeds up to 1600 MHz
  - UART Interface
  - I^2C Interface
  - Sixteen GPIO pins
  - SPI Interface
  - Semaphore Module
  - Eight 64-Bit Timers
  - Three On-Chip PLLs
- Commercial Temperature:
  - 0°C to 100°C
- Extended Temperature:
  - - 40°C to 100°C

1.1 KeyStone Architecture

TI's KeyStone Multicore Architecture provides a high-performance structure for integrating RISC and DSP cores with application-specific coprocessors and I/O. KeyStone is the first of its kind in that it provides adequate internal bandwidth for nonblocking access to all processing cores, peripherals, coprocessors, and I/O. This is achieved with four main hardware elements: Multicore Navigator, TeraNet, Multicore Shared Memory Controller, and HyperLink.

Multicore Navigator is an innovative packet-based manager that controls 8192 queues. When tasks are allocated to the queues, Multicore Navigator provides hardware-accelerated dispatch that directs tasks to the appropriate available hardware. The packet-based system on a chip (SoC) uses the 2-Tbps capacity of the TeraNet switched central resource to move packets. The Multicore Shared Memory Controller enables processing cores to access shared memory directly without drawing from the TeraNet's capacity, so packet movement cannot be blocked by memory access.

HyperLink provides a 50-GBaud chip-level interconnect that allows SoCs to work in tandem. Its low-protocol overhead and high throughput make Hyperlink an ideal interface for chip-to-chip interconnections. Working with Multicore Navigator, HyperLink dispatches tasks to tandem devices transparently and executes tasks as if they are running on local resources.

1.2 Device Description

The TMS320C6670 Communications Infrastructure KeyStone SoC is a member of the C66xx SoC family based on TI's new KeyStone Multicore SoC Architecture designed specifically for high performance wireless infrastructure applications. The C6670 provides a very high performance macro basestation platform for developing all wireless standards including WCDMA/HSPA/HSPA+, TD-SCDMA, GSM, TDD-LTE, FDD-LTE, and WiMAX. Even with aggregate data rates for 20-MHz LTE systems above 400 Mbps per sector, the C6670 can support two sectors running at full rate. The C6670 also sets a new standard for clock speed with operating frequencies up to 1.2 GHz.

TI's SoC architecture provides a programmable platform integrating various subsystems (C66x CorePacs, IP network, radio layers 1 and 2, and transport processing) and uses a queue-based communication system that allows the SoC resources to operate efficiently and seamlessly. This unique SoC architecture also includes a TeraNet Switch that enables the wide mix of system elements, from programmable cores to dedicated coprocessors and high speed IO, to each operate at maximum efficiency with no blocking or stalling.

TI's new C66x core launches a new era of DSP technology by combining fixed-point and floating-point computational capability in the processor without sacrificing speed, size, or power consumption. The raw computational performance is an industry-leading 32 GMACS/core and 16 Gflops/core (@ 1.2 GHz operating frequency). The C66x is also 100% backward compatible with software for C64x+ devices. The C66x core incorporates 90 new instructions targeted for floating-point (FPi) and vector-math-oriented (VPi) processing. These enhancements yield tremendous performance improvements in multi-antenna 4.8G signal processing for algorithms like MIMO and beamforming.

The C6670 contains many wireless basestation coprocessors to offload the bulk of the processing demands of layer 1 and layer 2 base station processing. This keeps the cores free for receiver algorithms and other differentiating functions. The SoC contains multiple copies of key coprocessors such as the FFTC and TCP3d. A key coprocessor for enabling high data rates is the bit rate coprocessor (BCP), which handles the entire downlink bit processing chain and much of the receive bit processing.The architectural elements of the SoC (Multicore Navigator) ensure that all the bits are processed without any CPU intervention or overhead, allowing the system to make optimal use of its resources.

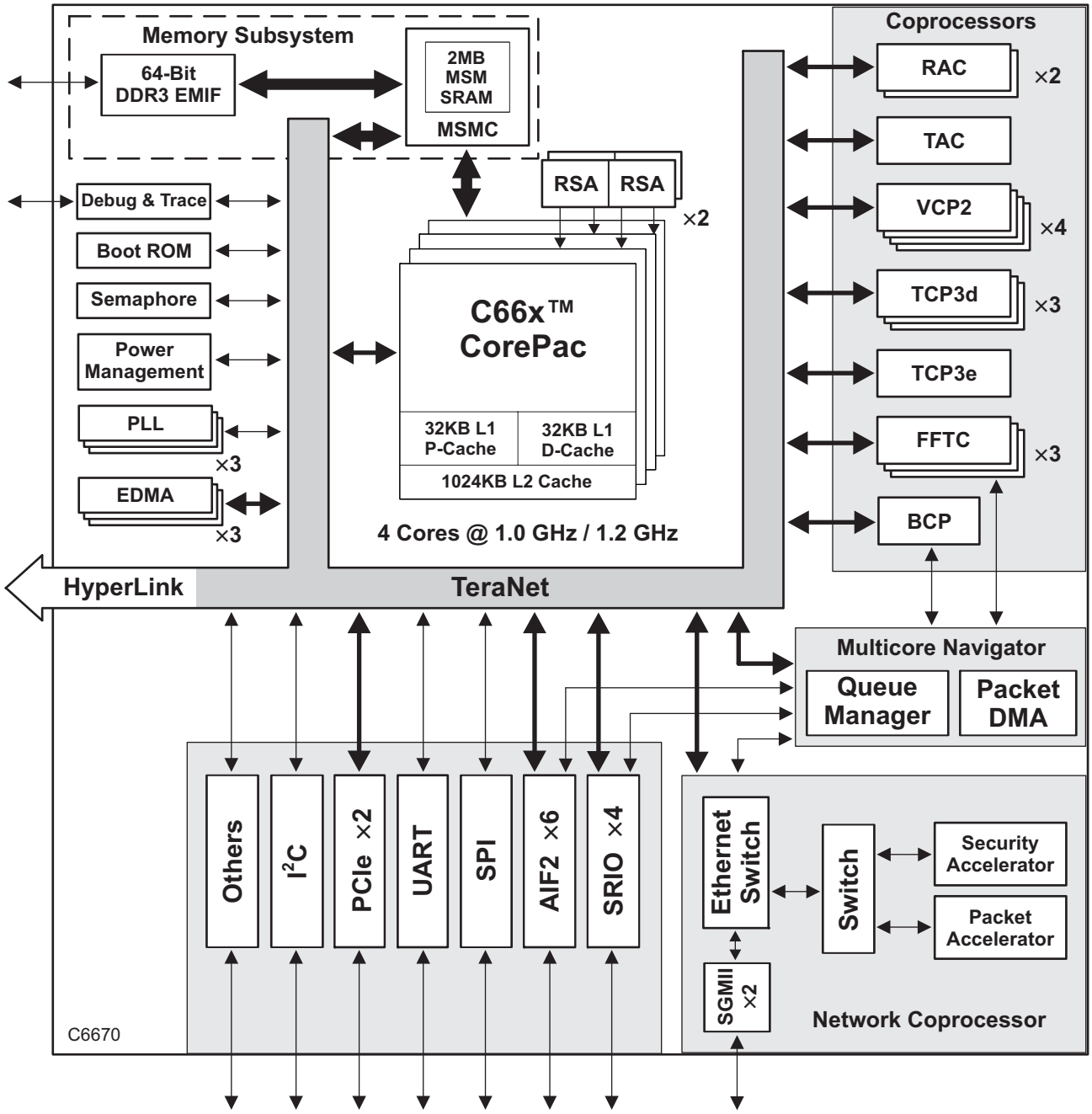
TI's scalable multicore SoC architecture solutions provide developers with a range of software- and hardware-compatible devices to minimize development time and maximize reuse across all base station platforms from Femto to Macro.

The C6670 device has a complete set of development tools that includes: a C compiler, an assembly optimizer to simplify programming and scheduling, and a Windows debugger interface for visibility into source code execution.

1.3 Functional Block Diagram

Figure 1-1 shows the functional block diagram of the TMS320C6670 device.

Figure 1-1 Functional Block Diagram



2 Device Overview

2.1 Device Characteristics

Table 2-1 provides an overview of the TMS320C6670 SoC. The table shows the significant features of the device, including the capacity of on-chip RAM, the peripherals, the CPU frequency, and the package type with pin count.

Table 2-1 Characteristics of the C6670 SoC (Part 1 of 2)

Hardware Features		TMS320C6670
Peripherals	DDR3 memory controller (64-bit bus width) [1.5-V I/O] (clock source = DDRREFCLKN|P)	1
	EDMA3 (16 independent channels) [CPU/2 clock rate]	1
	EDMA3 (64 independent channels) [CPU/3 clock rate]	2
	High-speed 1×/2×/4× Serial RapidIO port (4 lanes)	1
	Second-generation Antenna Interface (AIF2)	1
	I^2C	1
	SPI	1
	PCIe (2 lanes)	1
	UART	1
	10/100/1000 Ethernet	2
	Management Data Input/Output (MDIO)	1
	64-Bit Timers (configurable) (internal clock source = CPU/6 clock frequency)	Eight 64-bit ***or*** Sixteen 32-bit
	General-Purpose Input/Output Port (GPIO)	16
Encoder/Decoder Coprocessors	VCP2 (clock source = CPU/3 clock frequency)	4
	TCP3d (clock source = CPU/2 clock frequency)	3
	TCP3e (clock source = CPU/3 clock frequency)	1
	FFTC (clock source = CPU/3 clock frequency)	3
	BCP (clock source = CPU/3 clock frequency)	1
Accelerators	Receive Accelerator (RAC)	2
	Transmit Accelerator (TAC)	1
	Rake/Search Accelerator (RSA)	4
	Packet Accelerator (PA)	1
	Security Accelerator (1) (SA)	1
On-Chip Memory	Size (Bytes)	6528K
	Organization	128KB L1 Program Memory Controller [SRAM/Cache] 128KB L1 Data Memory Controller [SRAM/Cache] 4096KB L2 Unified Memory/Cache 2048KB MSM SRAM 128KB L3 ROM
C66x CorePac Revision ID	CorePac Revision ID Register (address location: 0181 2000h)	See Section 5.5 "CorePac Revision" on page 104.
JTAG BSDL_ID	JTAGID register (address location: 0x02620018)	See Section 3.3.3 "JTAG ID (JTAGID) Register Description" on page 73
Frequency	MHz	1200 (1.2 GHz) [-1200] 1000 (1.0 GHz) [-1000]
Cycle Time	ns	0.83 ns [-1200] 1 ns [-1000]
Voltage	Core (V)	SmartReflex variable supply
	I/O (V)	1.0 V, 1.5 V, and 1.8 V

Table 2-1 Characteristics of the C6670 SoC (Part 2 of 2)

Hardware Features		TMS320C6670
BGA Package	24 mm × 24 mm	841-Pin Flip-Chip Plastic BGA (CYP)
Process Technology	μm	0.040 μm
Product Status (2)	Product Preview (PP), Advance Information (AI), or Production Data (PD)	PD
End of Table 2-1		

1 The Security Accelerator function is subject to export control and will be enabled *only* for approved device shipments.
2 PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

The C66x Central Processing Unit (CPU) extends the performance of the C64x+ and C674x CPUs through enhancements and new features. Many of the new features target increased performance for vector processing. The C64x+ and C674x CPUs support 2-way SIMD operations for 16-bit data and 4-way SIMD operations for 8-bit data. On the C66x CPU, the vector processing capability is improved by extending the width of the SIMD instructions. C66x CPUs can execute instructions that operate on 128-bit vectors. For example the QMPY32 instruction is able to perform the element-to-element multiplication between two vectors of four 32-bit data each. The C66x CPU also supports SIMD for floating-point operations. Improved vector processing capability (each instruction can process multiple data in parallel) combined with the natural instruction-level parallelism of C6000 architecture (e.g execution of up to 8 instructions per cycle) results in a very high level of parallelism that can be exploited by DSP programmers through the use of TI's optimized C/C++ compiler.

The C66x CPU consists of eight functional units, two register files, and two data paths as shown in Figure 2-1. The two general-purpose register files (A and B) each contain thirty-two 32-bit registers for a total of 64 registers. The general-purpose registers can be used for data or can be data address pointers. The data types supported include packed 8-bit data, packed 16-bit data, 32-bit data, 40-bit data, and 64-bit data. Multiplies also support 128-bit data. 40-bit-long or 64-bit-long values are stored in register pairs, with the 32 LSBs of data placed in an even register and the remaining 8 or 32 MSBs in the next upper register (which is always an odd-numbered register). 128-bit data values are stored in register quadruplets, with the 32 LSBs of data placed in a register that is a multiple of 4 and the remaining 96 MSBs in the next 3 upper registers.

The eight functional units (.M1, .L1, .D1, .S1, .M2, .L2, .D2, and .S2) are each capable of executing one instruction every clock cycle. The .M functional units perform all multiply operations. The .S and .L units perform a general set of arithmetic, logical, and branch functions. The .D units primarily load data from memory to the register file and store results from the register file into memory.

Each C66x .M unit can perform one of the following fixed-point operations each clock cycle: four 32 × 32 bit multiplies, sixteen 16 × 16 bit multiplies, four 16 × 32 bit multiplies, four 8 × 8 bit multiplies, four 8 × 8 bit multiplies with add operations, and four 16 × 16 multiplies with add/subtract capabilities. There is also support for Galois field multiplication for 8-bit and 32-bit data. Many communications algorithms such as FFTs and modems require complex multiplication. Each C66x .M unit can perform one 16 × 16 bit complex multiply with or without rounding capabilities, two 16 × 16 bit complex multiplies with rounding capability, and a 32 × 32 bit complex multiply with rounding capability. The C66x can also perform two 16 × 16 bit and one 32 × 32 bit complex multiply instructions that multiply a complex number with a complex conjugate of another number with rounding capability. Communication signal processing also requires an extensive use of matrix operations. Each C66x .M unit is capable of multiplying a [1 × 2] complex vector by a [2 × 2] complex matrix per cycle with or without rounding capability. A version also exists allowing multiplication of the conjugate of a [1 × 2] vector with a [2 × 2] complex matrix.

Each C66x .M unit also includes IEEE floating-point multiplication operations from the C674x CPU. This includes one single-precision multiply each cycle and one double precision multiply every 4 cycles. There is also a mixed-precision multiply that allows multiplication of a single-precision value by a double-precision value and an operation allowing multiplication of two single-precision numbers resulting in a double-precision number. The

C66x CPU improves the performance over the C674x double-precision multiplies by adding a instruction allowing one double-precision multiply per cycle and also reduces the number of delay slots from ten to four. Each C66x .M unit can also perform one the following floating-point operations each clock cycle: one, two, or four single-precision multiplies or a complex single-precision multiply.

The .L and .S units can now support up to 64-bit operands. This allows for new versions of many of the arithmetic, logical, and data packing instructions to allow for more parallel operations per cycle. Additional instructions were added yielding performance enhancements of the floating point addition and subtraction instructions, including the ability to perform one double-precision addition or subtraction per cycle. Conversion to/from integer and single-precision values can now be done on both .L and .S units on the C66x. Also, by taking advantage of the larger operands, instructions were also added to double the number of these conversions that can be done. The .L unit also has additional instructions for logical AND and OR instructions, as well as 90 degree or 270 degree rotation of complex numbers (up to two per cycle). Instructions have also been added that allow for computing the conjugate of a complex number.

The MFENCE instruction is a new instruction introduced with the C66x DSP. This instruction creates a CPU stall until the completion of all the CPU-triggered memory transactions, including:

- Cache line fills
- Writes from L1D to L2 or from the CorePac to MSMC and/or other system endpoints
- Victim write backs
- Block or global coherence operations
- Cache mode changes
- Outstanding XMC prefetch requests

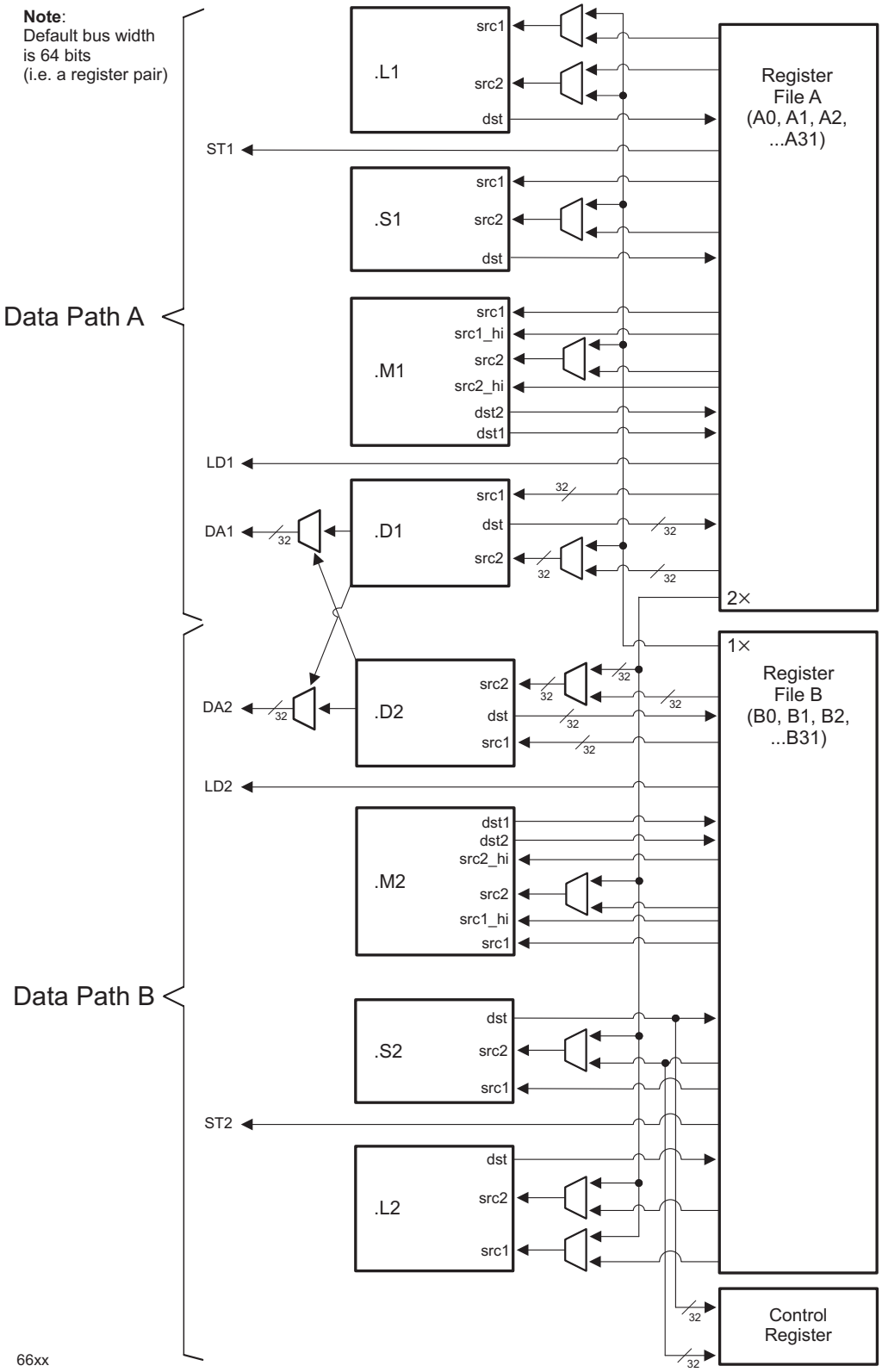
This is useful as a simple mechanism for programs to wait for these requests to reach their endpoint. It also provides ordering guarantees for writes arriving at a single endpoint via multiple paths, multiprocessor algorithms that depend on ordering, and manual coherence operations.

For more details on the C66x CPU and its enhancements over the C64x+ and C674x architectures, see the following documents (2.9 "Related Documentation from Texas Instruments" on page 66):

- *C66x CPU and Instruction Set Reference Guide*
- *C66x DSP Cache User Guide*
- *C66x CorePac User Guide*

Figure 2-1 shows the DSP core functional units and data paths.

Figure 2-1 CPU (DSP Core) Data Paths



2.2 Memory Map Summary

Table 2-2 shows the memory map address ranges of the TMS320C6670 device.

Table 2-2 Memory Map Summary (Part 1 of 9)

Logical 32 bit Address		Physical 36 bit Address			
Start	End	Start	End	Bytes	Description
0000 0000	007F FFFF	0 0000 0000	0 007F FFFF	8M	Reserved
0080 0000	008F FFFF	0 0080 0000	0 008F FFFF	1M	L2 SRAM
0090 0000	00DF FFFF	0 0090 0000	0 00DF FFFF	5M	Reserved
00E00000	00E0 7FFF	0 00E00000	0 00E0 7FFF	32K	L1P SRAM
00E08000	00EF FFFF	0 00E08000	0 00EF FFFF	1M-32K	Reserved
00F00000	00F0 7FFF	0 00F00000	0 00F0 7FFF	32K	L1D SRAM
00F08000	00FF FFFF	0 00F08000	0 00FF FFFF	1M-32K	Reserved
0100 0000	01BF FFFF	0 0100 0000	0 01BF FFFF	12 M	C66x CorePac registers
01C0 0000	01CF FFFF	0 01C0 0000	0 01CF FFFF	1M	Reserved
01D0 0000	01D0 007F	0 01D0 0000	0 01D0 007F	128	Tracer 0
01D0 0080	01D0 7FFF	0 01D0 0080	0 01D0 7FFF	32K-128	Reserved
01D0 8000	01D0 807F	0 01D0 8000	0 01D0 807F	128	Tracer 1
01D0 8080	01D0 FFFF	0 01D0 8080	0 01D0 FFFF	32K-128	Reserved
01D1 0000	01D1 007F	0 01D1 0000	0 01D1 007F	128	Tracer 2
01D1 0080	01D1 7FFF	0 01D1 0080	0 01D1 7FFF	32K-128	Reserved
01D1 8000	01D1 807F	0 01D1 8000	0 01D1 807F	128	Tracer 3
01D1 8080	01D1 FFFF	0 01D1 8080	0 01D1 FFFF	32K-128	Reserved
01D2 0000	01D2 007F	0 01D2 0000	0 01D2 007F	128	Tracer 4
01D2 0080	01D2 7FFF	0 01D2 0080	0 01D2 7FFF	32K-128	Reserved
01D2 8000	01D2 807F	0 01D2 8000	0 01D2 807F	128	Tracer 5
01D2 8080	01D2 FFFF	0 01D2 8080	0 01D2 FFFF	32K-128	Reserved
01D3 0000	01D3 007F	0 01D3 0000	0 01D3 007F	128	Tracer 6
01D3 0080	01D3 7FFF	0 01D3 0080	0 01D3 7FFF	32K-128	Reserved
01D3 8000	01D3 807F	0 01D3 8000	0 01D3 807F	128	Tracer 7
01D3 8080	01D3 FFFF	0 01D3 8080	0 01D3 FFFF	32K-128	Reserved
01D4 0000	01D4 007F	0 01D4 0000	0 01D4 007F	128	Tracer 8
01D4 0080	01D4 7FFF	0 01D4 0080	0 01D4 7FFF	32K-128	Reserved
01D4 8000	01D4 807F	0 01D4 8000	0 01D4 807F	128	Tracer 9
01D4 8080	01D4 FFFF	0 01D4 8080	0 01D4 FFFF	32K-128	Reserved
01D5 0000	01D5 007F	0 01D5 0000	0 01D5 007F	128	Tracer 10
01D5 0080	01D5 7FFF	0 01D5 0080	0 01D5 7FFF	32K-128	Reserved
01D5 8000	01D5 807F	0 01D5 8000	0 01D5 807F	128	Tracer 11
01D5 8080	01D5 FFFF	0 01D5 8080	0 01D5 FFFF	32K-128	Reserved
01D6 0000	01D6 007F	0 01D6 0000	0 01D6 007F	128	Tracer 12
01D6 0080	01D6 7FFF	0 01D6 0080	0 01D6 7FFF	32K-128	Reserved
01D6 8000	01D6 807F	0 01D6 8000	0 01D6 807F	128	Tracer 13
01D6 8080	01D6 FFFF	0 01D6 8080	0 01D6 FFFF	32K-128	Reserved
01D7 0000	01D7 007F	0 01D7 0000	0 01D7 007F	128	Tracer 14
01D7 0080	01D7 7FFF	0 01D7 0080	0 01D7 7FFF	32K-128	Reserved
01D7 8000	01D7 807F	0 01D7 8000	0 01D7 807F	128	Tracer 15

Table 2-2 Memory Map Summary (Part 2 of 9)

Logical 32 bit Address		Physical 36 bit Address			
Start	End	Start	End	Bytes	Description
01D7 8080	01D7 FFFF	0 01D7 8080	0 01D7 FFFF	32K-128	Reserved
01D8 0000	01D8 007F	0 01D8 0000	0 01D8 007F	128	Reserved
01D8 0080	01D8 7FFF	0 01D8 0080	0 01D8 7FFF	32K-128	Reserved
01D8 8000	01DF FFFF	0 01D8 8000	0 01DF FFFF	480K	Reserved
01E0 0000	01E3 FFFF	0 01E0 0000	0 01E3 FFFF	256K	Reserved
01E4 0000	01E7 FFFF	0 01E4 0000	0 01E7 FFFF	256K	Reserved
01E8 0000	01EB FFFF	0 01E8 0000	0 01EB FFFF	256K	Reserved
01EC 0000	01EF FFFF	0 01EC 0000	0 01EF FFFF	256K	Reserved
01F0 0000	01F7 FFFF	0 01F0 0000	0 01F7 FFFF	512K	AIF2 control
01F8 0000	01F8 FFFF	0 01F8 0000	0 01F8 FFFF	64K	RAC_B - FEI control
01F9 0000	01F9 FFFF	0 01F9 0000	0 01F9 FFFF	64K	RAC_B - BEI control
01FA 0000	01FB FFFF	0 01FA 0000	0 01FB FFFF	128K	RAC_B - GCCP 0 control
01FC 0000	01FD FFFF	0 01FC 0000	0 01FD FFFF	128K	RAC_B - GCCP 1 control
01FE 0000	01FF FFFF	0 01FE 0000	0 01FF FFFF	128K	Reserved
0200 0000	020F FFFF	0 0200 0000	0 020F FFFF	1M	Network Coprocessor (Packet Accelerator, Gigabit Ethernet Switch subsystem, and Security Accelerator)
0210 0000	0210 FFFF	0 0210 0000	0 0210 FFFF	64K	RAC_A - FEI control
0211 0000	0211 FFFF	0 0211 0000	0 0211 FFFF	64K	RAC_A - BEI control
0212 0000	0213 FFFF	0 0212 0000	0 0213 FFFF	128K	RAC_A - GCCP 0 control
0214 0000	0215 FFFF	0 0214 0000	0 0215 FFFF	128K	RAC_A - GCCP 1 control
0216 0000	0217 FFFF	0 0216 0000	0 0217 FFFF	128K	Reserved
0218 0000	0218 7FFF	0 0218 0000	0 0218 7FFF	32K	TAC - FEI control
0218 8000	0218 FFFF	0 0218 8000	0 0218 FFFF	32K	TAC- BEI control
0219 0000	0219 FFFF	0 0219 0000	0 0219 FFFF	64K	TAC - SGCCP 0 control
021A 0000	021A FFFF	0 021A 0000	0 021A FFFF	64K	TAC - SGCCP 1 control
021B 0000	021B FFFF	0 021B 0000	0 021B FFFF	64K	Reserved
021C 0000	021C 03FF	0 021C 0000	0 021C 03FF	1K	TCP3d-A
021C 0400	021C 7FFF	0 021C 0400	0 021C 7FFF	31K	Reserved
021C 8000	021C 83FF	0 021C 8000	0 021C 83FF	1K	TCP3d-B
021C 8400	021C FFFF	0 021C 8400	0 021C FFFF	31K	Reserved
021D 0000	021D 00FF	0 021D 0000	0 021D 00FF	256	VCP2_A
021D 0100	021D 3FFF	0 021D 0100	0 021D 3FFF	16K	Reserved
021D 4000	021D 40FF	0 021D 4000	0 021D 40FF	256	VCP2_B
021D 4100	021D 7FFF	0 021D 4100	0 021D 7FFF	16K	Reserved
021D 8000	021D 80FF	0 021D 8000	0 021D 80FF	256	VCP2_C
021D 8100	021D BFFF	0 021D 8100	0 021D BFFF	16K	Reserved
021D C000	021D C0FF	0 021D C000	0 021D C0FF	256	VCP2_D
021D C100	021D FFFF	0 021D C100	0 021D FFFF	16K	Reserved
021E 0000	021E 0FFF	0 021E 0000	0 021E 0FFF	4K	TCP3e
021E 1000	021E FFFF	0 021E 1000	0 021E FFFF	60K	Reserved
021F 0000	021F 07FF	0 021F 0000	0 021F 07FF	2K	FFTC-A configuration
021F 0800	021F 3FFF	0 021F 0800	0 021F 3FFF	14K	Reserved
021F 4000	021F 47FF	0 021F 4000	0 021F 47FF	2K	FFTC-B configuration

Table 2-2 Memory Map Summary (Part 3 of 9)

Logical 32 bit Address		Physical 36 bit Address			
Start	End	Start	End	Bytes	Description
021F 4800	021F FFFF	0 021F 4800	0 021F FFFF	46K	Reserved
0220 0000	0220 007F	0 0220 0000	0 0220 007F	128	Timer0
0220 0080	0220 FFFF	0 0220 0080	0 0220 FFFF	64K-128	Reserved
0221 0000	0221 007F	0 0221 0000	0 0221 007F	128	Timer1
0221 0080	0221 FFFF	0 0221 0080	0 0221 FFFF	64K-128	Reserved
0222 0000	0222 007F	0 0222 0000	0 0222 007F	128	Timer2
0222 0080	0222 FFFF	0 0222 0080	0 0222 FFFF	64K-128	Reserved
0223 0000	0223 007F	0 0223 0000	0 0223 007F	128	Timer3
0223 0080	0223 FFFF	0 0223 0080	0 0223 FFFF	64K-128	Reserved
0224 0000	0224 007F	0 0224 0000	0 0224 007F	128	Timer4
0224 0080	0224 FFFF	0 0224 0080	0 0224 FFFF	64K-128	Reserved
0225 0000	0225 007F	0 0225 0000	0 0225 007F	128	Timer5
0225 0080	0225 FFFF	0 0225 0080	0 0225 FFFF	64K-128	Reserved
0226 0000	0226 007F	0 0226 0000	0 0226 007F	128	Timer6
0226 0080	0226 FFFF	0 0226 0080	0 0226 FFFF	64K-128	Reserved
0227 0000	0227 007F	0 0227 0000	0 0227 007F	128	Timer7
0227 0080	0227 FFFF	0 0227 0080	0 0227 FFFF	64K-128	Reserved
0228 0000	0228 007F	0 0228 0000	0 0228 007F	128	Reserved
0228 0080	0228 FFFF	0 0228 0080	0 0228 FFFF	64K-128	Reserved
0229 0000	0229 007F	0 0229 0000	0 0229 007F	128	Reserved
0229 0080	0229 FFFF	0 0229 0080	0 0229 FFFF	64K-128	Reserved
022A 0000	022A 007F	0 022A 0000	0 022A 007F	128	Reserved
022A 0080	022A FFFF	0 022A 0080	0 022A FFFF	64K-128	Reserved
022B 0000	022B 007F	0 022B 0000	0 022B 007F	128	Reserved
022B 0080	022B FFFF	0 022B 0080	0 022B FFFF	64K-128	Reserved
022C 0000	022C 007F	0 022C 0000	0 022C 007F	128	Reserved
022C 0080	022C FFFF	0 022C 0080	0 022C FFFF	64K-128	Reserved
022D 0000	022D 007F	0 022D 0000	0 022D 007F	128	Reserved
022D 0080	022D FFFF	0 022D 0080	0 022D FFFF	64K-128	Reserved
022E 0000	022E 007F	0 022E 0000	0 022E 007F	128	Reserved
022E 0080	022E FFFF	0 022E 0080	0 022E FFFF	64K-128	Reserved
022F 0000	022F 007F	0 022F 0000	0 022F 007F	128	Reserved
022F 0080	022F FFFF	0 022F 0080	0 022F FFFF	64K-128	Reserved
0230 0000	0230 FFFF	0 0230 0000	0 0230 FFFF	64K	Reserved
0231 0000	0231 01FF	0 0231 0000	0 0231 01FF	512	PLL controller
0231 0200	0231 FFFF	0 0231 0200	0 0231 FFFF	64K-512	Reserved
0232 0000	0232 00FF	0 0232 0000	0 0232 00FF	256	GPIO
0232 0100	0232 FFFF	0 0232 0100	0 0232 FFFF	64K-256	Reserved
0233 0000	0233 03FF	0 0233 0000	0 0233 03FF	1K	SmartReflex
0233 0400	0233 FFFF	0 0233 0400	0 0233 FFFF	63K	Reserved
0234 0000	0234 FFFF	0 0234 0000	0 0234 FFFF	64K	Reserved
0235 0000	0235 0FFF	0 0235 0000	0 0235 0FFF	4K	Power Sleep Controller (PSC)
0235 1000	0235 FFFF	0 0235 1000	0 0235 FFFF	64K-4K	Reserved

Table 2-2 Memory Map Summary (Part 4 of 9)

Logical 32 bit Address		Physical 36 bit Address			
Start	End	Start	End	Bytes	Description
0236 0000	0236 03FF	0 0236 0000	0 0236 03FF	1K	Memory Protection Unit (MPU) 0
0236 0400	0236 7FFF	0 0236 0400	0 0236 7FFF	31K	Reserved
0236 8000	0236 83FF	0 0236 8000	0 0236 83FF	1K	Memory Protection Unit (MPU) 1
0236 8400	0236 FFFF	0 0236 8400	0 0236 FFFF	31K	Reserved
0237 0000	0237 03FF	0 0237 0000	0 0237 03FF	1K	Memory Protection Unit (MPU) 2
0237 0400	0237 7FFF	0 0237 0400	0 0237 7FFF	31K	Reserved
0237 8000	0237 83FF	0 0237 8000	0 0237 83FF	1K	Memory Protection Unit (MPU) 3
0237 8400	0237 FFFF	0 0237 8400	0 0237 FFFF	31K	Reserved
0238 0000	0238 03FF	0 0238 0000	0 0238 03FF	1K	Memory Protection Unit (MPU) 4
0238 0400	023F FFFF	0 0238 0400	0 023F FFFF	511K	Reserved
0240 0000	0243 FFFF	0 0240 0000	0 0243 FFFF	256K	Reserved
0244 0000	0244 3FFF	0 0244 0000	0 0244 3FFF	16K	DSP trace formatter 0
0244 4000	0244 FFFF	0 0244 4000	0 0244 FFFF	48K	Reserved
0245 0000	0245 3FFF	0 0245 0000	0 0245 3FFF	16K	DSP trace formatter 1
0245 4000	0245 FFFF	0 0245 4000	0 0245 FFFF	48K	Reserved
0246 0000	0246 3FFF	0 0246 0000	0 0246 3FFF	16K	DSP trace formatter 2
0246 4000	0246 FFFF	0 0246 4000	0 0246 FFFF	48K	Reserved
0247 0000	0247 3FFF	0 0247 0000	0 0247 3FFF	16K	DSP trace formatter 3
0247 4000	0247 FFFF	0 0247 4000	0 0247 FFFF	48K	Reserved
0248 0000	0248 3FFF	0 0248 0000	0 0248 3FFF	16K	Reserved
0248 4000	0248 FFFF	0 0248 4000	0 0248 FFFF	48K	Reserved
0249 0000	0249 3FFF	0 0249 0000	0 0249 3FFF	16K	Reserved
0249 4000	0249 FFFF	0 0249 4000	0 0249 FFFF	48K	Reserved
024A 0000	024A 3FFF	0 024A 0000	0 024A 3FFF	16K	Reserved
024A 4000	024A FFFF	0 024A 4000	0 024A FFFF	48K	Reserved
024B 0000	024B 3FFF	0 024B 0000	0 024B 3FFF	16K	Reserved
024B 4000	024B FFFF	0 024B 4000	0 024B FFFF	48K	Reserved
024C 0000	024C 01FF	0 024C 0000	0 024C 01FF	512	Reserved
024C 0200	024C 03FF	0 024C 0200	0 024C 03FF	1K-512	Reserved
024C 0400	024C 07FF	0 024C 0400	0 024C 07FF	1K	Reserved
024C 0800	024C FFFF	0 024C 0800	0 024C FFFF	62K	Reserved
024D 0000	024F FFFF	0 024D 0000	0 024F FFFF	192K	Reserved
0250 0000	0250 007F	0 0250 0000	0 0250 007F	128	Reserved
0250 0080	0250 7FFF	0 0250 0080	0 0250 7FFF	32K-128	Reserved
0250 8000	0250 FFFF	0 0250 8000	0 0250 FFFF	32K	Reserved
0251 0000	0251 FFFF	0 0251 0000	0 0251 FFFF	64K	Reserved
0252 0000	0252 03FF	0 0252 0000	0 0252 03FF	1K	Reserved
0252 0400	0252 FFFF	0 0252 0400	0 0252 FFFF	64K-1K	Reserved
0253 0000	0253 007F	0 0253 0000	0 0253 007F	128	I^2C data & control
0253 0080	0253 FFFF	0 0253 0080	0 0253 FFFF	64K-128	Reserved
0254 0000	0254 003F	0 0254 0000	0 0254 003F	64	UART
0254 0400	0254 FFFF	0 0254 0400	0 0254 FFFF	64K-64	Reserved
0255 0000	0257 FFFF	0 0255 0000	0 0257 FFFF	192K	Reserved

Table 2-2 Memory Map Summary (Part 5 of 9)

Logical 32 bit Address		Physical 36 bit Address			
Start	End	Start	End	Bytes	Description
0258 0000	025B FFFF	0 0258 0000	0 025B FFFF	256K	Reserved
025C 0000	025F FFFF	0 025C 0000	0 025F FFFF	256K	Reserved
0260 0000	0260 1FFF	0 0260 0000	0 0260 1FFF	8K	Secondary Chip Interrupt Contoller CIC 0
0260 2000	0260 3FFF	0 0260 2000	0 0260 3FFF	8K	Reserved
0260 4000	0260 5FFF	0 0260 4000	0 0260 5FFF	8K	Secondary Chip Interrupt Contoller CIC 1
0260 6000	0260 7FFF	0 0260 6000	0 0260 7FFF	8K	Reserved
0260 8000	0260 9FFF	0 0260 8000	0 0260 9FFF	8K	Secondary Chip Interrupt Contoller CIC 2
0260 A000	0260 BFFF	0 0260 A000	0 0260 BFFF	8K	Reserved
0260 C000	0260 DFFF	0 0260 C000	0 0260 DFFF	8K	Reserved
0260 E000	0260 FFFF	0 0260 E000	0 0260 FFFF	8K	Reserved
0261 0000	0261 FFFF	0 0261 0000	0 0261 FFFF	64K	Reserved
0262 0000	0262 03FF	0 0262 0000	0 0262 03FF	1K	Chip-level registers
0262 0400	0262 FFFF	0 0262 0400	0 0262 FFFF	63K	Reserved
0263 0000	0263 FFFF	0 0263 0000	0 0263 FFFF	64K	Reserved
0264 0000	0264 07FF	0 0264 0000	0 0264 07FF	2K	Semaphore
0264 0800	0264 FFFF	0 0264 0800	0 0264 FFFF	64K-2K	Reserved
0265 0000	026F FFFF	0 0265 0000	0 026F FFFF	704K	Reserved
0270 0000	0270 7FFF	0 0270 0000	0 0270 7FFF	32K	EDMA3 channel controller EDMA3CC0
0270 8000	0271 FFFF	0 0270 8000	0 0271 FFFF	96K	Reserved
0272 0000	0272 7FFF	0 0272 0000	0 0272 7FFF	32K	EDMA3 channel controller EDMA3CC1
0272 8000	0273 FFFF	0 0272 8000	0 0273 FFFF	96K	Reserved
02740000	0274 7FFF	0 02740000	0 0274 7FFF	32K	EDMA3 channel controller EDMA3CC2
0274 8000	0275 FFFF	0 0274 8000	0 0275 FFFF	96K	Reserved
0276 0000	0276 03FF	0 0276 0000	0 0276 03FF	1K	EDMA3CC0 transfer controller EDMA3TC0
0276 0400	0276 7FFF	0 0276 0400	0 0276 7FFF	31K	Reserved
0276 8000	0276 83FF	0 0276 8000	0 0276 83FF	1K	EDMA3CC0 transfer controller EDMA3TC1
0276 8400	0276 FFFF	0 0276 8400	0 0276 FFFF	31K	Reserved
0277 0000	0277 03FF	0 0277 0000	0 0277 03FF	1K	EDMA3CC1 transfer controller EDMA3TC0
0277 0400	0277 7FFF	0 0277 0400	0 0277 7FFF	31K	Reserved
0277 8000	0277 83FF	0 0277 8000	0 0277 83FF	1K	EDMA3CC1 transfer controller EDMA3TC1
0278 0400	0277 FFFF	0 0278 0400	0 0277 FFFF	31K	Reserved
0278 0000	0278 03FF	0 0278 0000	0 0278 03FF	1K	EDMA3CC1 transfer controller EDMA3TC2
0278 0400	0278 7FFF	0 0278 0400	0 0278 7FFF	31K	Reserved
0278 8000	0278 83FF	0 0278 8000	0 0278 83FF	1K	EDMA3CC1 transfer controller EDMA3TC3
0278 8400	0278 FFFF	0 0278 8400	0 0278 FFFF	31K	Reserved
0279 0000	0279 03FF	0 0279 0000	0 0279 03FF	1K	EDMA3CC2 transfer controllerEDMA3TC0
0279 0400	0279 7FFF	0 0279 0400	0 0279 7FFF	31K	Reserved
0279 8000	0279 83FF	0 0279 8000	0 0279 83FF	1K	EDMA3CC2 transfer controller EDMA3TC1
0279 8400	0279 FFFF	0 0279 8400	0 0279 FFFF	31K	Reserved
027A 0000	027A 03FF	0 027A 0000	0 027A 03FF	1K	EDMA3CC2 transfer controllerEDMA3TC2
027A 0400	027A 7FFF	0 027A 0400	0 027A 7FFF	31K	Reserved
027A 8000	027A 83FF	0 027A 8000	0 027A 83FF	1K	EDMA3CC2 transfer controller EDMA3TC3
027A 8400	027A FFFF	0 027A 8400	0 027A FFFF	31K	Reserved

Table 2-2 Memory Map Summary (Part 6 of 9)

Logical 32 bit Address		Physical 36 bit Address			
Start	End	Start	End	Bytes	Description
027B 0000	027B FFFF	0 027B 0000	0 027B FFFF	64K	Reserved
027C 0000	027C FFFF	0 027C 0000	0 027C FFFF	64K	Reserved
027D 0000	027D 3FFF	0 027D 0000	0 027D 3FFF	16K	TI embedded trace buffer (TETB) - CorePac0
027D 4000	027D FFFF	0 027D 4000	0 027D FFFF	48K	Reserved
027E 0000	027E 3FFF	0 027E 0000	0 027E 3FFF	16K	TI embedded trace buffer (TETB) - CorePac1
027E 4000	027E FFFF	0 027E 4000	0 027E FFFF	48K	Reserved
027F 0000	027F 3FFF	0 027F 0000	0 027F 3FFF	16K	TI embedded trace buffer (TETB) - CorePac2
027F 4000	027F FFFF	0 027F 4000	0 027F FFFF	48K	Reserved
0280 0000	0280 3FFF	0 0280 0000	0 0280 3FFF	16	TI embedded trace buffer (TETB) - CorePac3
0280 4000	0280 FFFF	0 0280 4000	0 0280 FFFF	48K	Reserved
0281 0000	0281 3FFF	0 0281 0000	0 0281 3FFF	16K	Reserved
0281 4000	0281 FFFF	0 0281 4000	0 0281 FFFF	48K	Reserved
0282 0000	0282 3FFF	0 0282 0000	0 0282 3FFF	16K	Reserved
0282 4000	0282 FFFF	0 0282 4000	0 0282 FFFF	48K	Reserved
0283 0000	0283 3FFF	0 0283 0000	0 0283 3FFF	16K	Reserved
0283 4000	0283 FFFF	0 0283 4000	0 0283 FFFF	48K	Reserved
0284 0000	0284 3FFF	0 0284 0000	0 0284 3FFF	16K	Reserved
0284 4000	0284 FFFF	0 0284 4000	0 0284 FFFF	48K	Reserved
0285 0000	0285 7FFF	0 0285 0000	0 0285 7FFF	32K	TI embedded trace buffer (TETB) - system
0285 8000	0285 FFFF	0 0285 8000	0 0285 FFFF	32K	Reserved
0286 0000	028F FFFF	0 0286 0000	0 028F FFFF	640K	Reserved
0290 0000	0290 0FFF	0 0290 0000	0 0290 0FFF	4K	Serial RapidIO (SRIO) configuration
0290 8000	029F FFFF	0 0290 8000	0 029F FFFF	1M-32K	Reserved
02A0 0000	02AF FFFF	0 02A0 0000	0 02AF FFFF	1M	Queue Manager subsystem configuration
02B0 0000	02BF FFFF	0 02B0 0000	0 02BF FFFF	1M	Reserved
02C0 0000	02FF FFFF	0 02C0 0000	0 02FF FFFF	4M	Reserved
03000 000	07FF FFFF	0 03000 000	0 07FF FFFF	80M	Reserved
0800 0000	0800 FFFF	0 0800 0000	0 0800 FFFF	64K	Extended Memory Controller (XMC) configuration
0801 0000	0BBF FFFF	0 0801 0000	0 0BBF FFFF	60M-64K	Reserved
0BC0 0000	0BCF FFFF	0 0BC0 0000	0 0BCF FFFF	1M	Multicore Shared Memory Controller (MSMC) config
0BD0 0000	0BFF FFFF	0 0BD0 0000	0 0BFF FFFF	3M	Reserved
0C00 0000	0C1F FFFF	0 0C00 0000	0 0C1F FFFF	2M	Multicore Shared Memory (MSM)
0C20 0000	0C3F FFFF	0 0C20 0000	0 0C3F FFFF	2M	Reserved
0C40 0000	0FFF FFFF	0 0C40 0000	0 0FFF FFFF	60M	Reserved
1000 0000	107F FFFF	0 1000 0000	0 107F FFFF	8M	Reserved
1080 0000	108F FFFF	0 1080 0000	0 108F FFFF	1M	CorePac0 L2 SRAM
1090 0000	10DF FFFF	0 1090 0000	0 10DF FFFF	5M	Reserved
10E0 0000	10E0 7FFF	0 10E0 0000	0 10E0 7FFF	32K	CorePac0 L1P SRAM
10E0 8000	10EF FFFF	0 10E0 8000	0 10EF FFFF	1M-32K	Reserved
10F0 0000	10F0 7FFF	0 10F0 0000	0 10F0 7FFF	32K	CorePac0 L1D SRAM
10F0 8000	117F FFFF	0 10F0 8000	0 117F FFFF	9M-32K	Reserved
1180 0000	118F FFFF	0 1180 0000	0 118F FFFF	1M	CorePac1 L2 SRAM
1190 0000	11DF FFFF	0 1190 0000	0 11DF FFFF	5M	Reserved

Table 2-2 Memory Map Summary (Part 7 of 9)

Logical 32 bit Address		Physical 36 bit Address			
Start	End	Start	End	Bytes	Description
11E0 0000	11E0 7FFF	0 11E0 0000	0 11E0 7FFF	32K	CorePac1 L1P SRAM
11E0 8000	11EF FFFF	0 11E0 8000	0 11EF FFFF	1M-32K	Reserved
11F0 0000	11F0 7FFF	0 11F0 0000	0 11F0 7FFF	32K	CorePac1 L1D SRAM
11F0 8000	127F FFFF	0 11F0 8000	0 127F FFFF	9M-32K	Reserved
1280 0000	128F FFFF	0 1280 0000	0 128F FFFF	1M	CorePac2 L2 SRAM
1290 0000	12DF FFFF	0 1290 0000	0 12DF FFFF	5M	Reserved
12E0 0000	12E0 7FFF	0 12E0 0000	0 12E0 7FFF	32K	CorePac2 L1P SRAM
12E0 8000	12EF FFFF	0 12E0 8000	0 12EF FFFF	1M-32K	Reserved
12F0 0000	12F0 7FFF	0 12F0 0000	0 12F0 7FFF	32K	CorePac2 L1D SRAM
12F0 8000	137F FFFF	0 12F0 8000	0 137F FFFF	9M-32K	Reserved
1380 0000	1388 FFFF	0 1380 0000	0 1388 FFFF	1M	CorePac3 L2 SRAM
1390 0000	13DF FFFF	0 1390 0000	0 13DF FFFF	5M	Reserved
13E0 0000	13E0 7FFF	0 13E0 0000	0 13E0 7FFF	32K	CorePac3 L1P SRAM
13E0 8000	13EF FFFF	0 13E0 8000	0 13EF FFFF	1M-32K	Reserved
13F0 0000	13F0 7FFF	0 13F0 0000	0 13F0 7FFF	32K	CorePac3 L1D SRAM
13F0 8000	147F FFFF	0 13F0 8000	0 147F FFFF	9M-32K	Reserved
1480 0000	1487 FFFF	0 1480 0000	0 1487 FFFF	512K	Reserved
1488 0000	148F FFFF	0 1488 0000	0 148F FFFF	512K	Reserved
1490 0000	14DF FFFF	0 1490 0000	0 14DF FFFF	5M	Reserved
14E0 0000	14E0 7FFF	0 14E0 0000	0 14E0 7FFF	32K	Reserved
14E0 8000	14EF FFFF	0 14E0 8000	0 14EF FFFF	1M-32K	Reserved
14F0 0000	14F0 7FFF	0 14F0 0000	0 14F0 7FFF	32K	Reserved
14F0 8000	157F FFFF	0 14F0 8000	0 157F FFFF	9M-32K	Reserved
1580 0000	1587 FFFF	0 1580 0000	0 1587 FFFF	512K	Reserved
1588 0000	158F FFFF	0 1588 0000	0 158F FFFF	512K	Reserved
1590 0000	15DF FFFF	0 1590 0000	0 15DF FFFF	5M	Reserved
15E0 0000	15E0 7FFF	0 15E0 0000	0 15E0 7FFF	32K	Reserved
15E0 8000	15EF FFFF	0 15E0 8000	0 15EF FFFF	1M-32K	Reserved
15F0 0000	15F0 7FFF	0 15F0 0000	0 15F0 7FFF	32K	Reserved
15F0 8000	167F FFFF	0 15F0 8000	0 167F FFFF	9M-32K	Reserved
1680 0000	1687 FFFF	0 1680 0000	0 1687 FFFF	512K	Reserved
1688 0000	168F FFFF	0 1688 0000	0 168F FFFF	512K	Reserved
1690 0000	16DF FFFF	0 1690 0000	0 16DF FFFF	5M	Reserved
16E0 0000	16E0 7FFF	0 16E0 0000	0 16E0 7FFF	32K	Reserved
16E0 8000	16EF FFFF	0 16E0 8000	0 16EF FFFF	1M-32K	Reserved
16F0 0000	16F0 7FFF	0 16F0 0000	0 16F0 7FFF	32K	Reserved
16F0 8000	177F FFFF	0 16F0 8000	0 177F FFFF	9M-32K	Reserved
1780 0000	1787 FFFF	0 1780 0000	0 1787 FFFF	512K	Reserved
1788 0000	178F FFFF	0 1788 0000	0 178F FFFF	512K	Reserved
1790 0000	17DF FFFF	0 1790 0000	0 17DF FFFF	5M	Reserved
17E0 0000	17E0 7FFF	0 17E0 0000	0 17E0 7FFF	32K	Reserved
17E0 8000	17EF FFFF	0 17E0 8000	0 17EF FFFF	1M-32K	Reserved
17F0 0000	17F0 7FFF	0 17F0 0000	0 17F0 7FFF	32K	Reserved

Table 2-2 Memory Map Summary (Part 8 of 9)

Logical 32 bit Address		Physical 36 bit Address			
Start	End	Start	End	Bytes	Description
17F0 8000	1FFF FFFF	0 17F0 8000	0 1FFF FFFF	129M-32K	Reserved
2000 0000	200F FFFF	0 2000 0000	0 200F FFFF	1M	System trace manager (STM) configuration
2010 0000	201F FFFF	0 2010 0000	0 201F FFFF	1M	Reserved
2020 0000	205F FFFF	0 2020 0000	0 205F FFFF	4M	RAC_B data
2060 0000	206F FFFF	0 2060 0000	0 206F FFFF	1M	TCP3d-B data
2070 0000	207F FFFF	0 2070 0000	0 207F FFFF	1M	Reserved
2080 0000	208F FFFF	0 2080 0000	0 208F FFFF	1M	TCP3d-A data
2090 0000	2090 1FFF	0 2090 0000	0 2090 1FFF	8K	TCP3e data write port
2090 2000	2090 3FFF	0 2090 2000	0 2090 3FFF	8K	TCP3e data read port
2090 4000	209F FFFF	0 2090 4000	0 209F FFFF	1M-16K	Reserved
20A0 0000	20A3 FFFF	0 20A0 0000	0 20A3 FFFF	256K	Reserved
20A4 0000	20A4 FFFF	0 20A4 0000	0 20A4 FFFF	64K	Reserved
20A5 0000	20AF FFFF	0 20A5 0000	0 20AF FFFF	704K	Reserved
20B0 0000	20B1 FFFF	0 20B0 0000	0 20B1 FFFF	128K	Boot ROM
20B2 0000	20BE FFFF	0 20B2 0000	0 20BE FFFF	832K	Reserved
20BF 0000	20BF 01FF	0 20BF 0000	0 20BF 01FF	512	SPI
20BF 0400	20BF FFFF	0 20BF 0400	0 20BF FFFF	63K	Reserved
20C0 0000	20C0 00FF	0 20C0 0000	0 20C0 00FF	256	Reserved
20C0 0100	20FF FFFF	0 20C0 0100	0 20FF FFFF	4M-256	Reserved
2100 0000	2100 01FF	1 0000 0000	1 0000 01FF	512	DDR3 EMIF configuration
2100 0200	213F FFFF	0 2100 0200	0 213F FFFF	4M-256	Reserved
2140 0000	2140 00FF	0 2140 0000	0 2140 00FF	256	HyperLink config
2140 0400	217F FFFF	0 2140 0400	0 217F FFFF	4M-1K	Reserved
2180 0000	2180 7FFF	0 2180 0000	0 2180 7FFF	32K	PCIe config
2180 8000	21BF FFFF	0 2180 8000	0 21BF FFFF	4M-32K	Reserved
21C0 0000	21FF FFFF	0 21C0 0000	0 21FF FFFF	4M	Reserved
2200 0000	229F FFFF	0 2200 0000	0 229F FFFF	10M	Reserved
22A0 0000	22A0 FFFF	0 22A0 0000	0 22A0 FFFF	64K	VCP2_A
22A1 0000	22AF FFFF	0 22A1 0000	0 22AF FFFF	1M-64K	Reserved
22B0 0000	22B0 FFFF	0 22B0 0000	0 22B0 FFFF	64K	VCP2_B
22B1 0000	22BF FFFF	0 22B1 0000	0 22BF FFFF	1M-64K	Reserved
22C0 0000	22C0 FFFF	0 22C0 0000	0 22C0 FFFF	64K	VCP2_C
22C1 0000	22CF FFFF	0 22C1 0000	0 22CF FFFF	1M-64K	Reserved
22D0 0000	22D0 FFFF	0 22D0 0000	0 22D0 FFFF	64K	VCP2_D
22D1 0000	22DF FFFF	0 22D1 0000	0 22DF FFFF	1M-64K	Reserved
22E0 0000	23FF FFFF	0 22E0 0000	0 23FF FFFF	18M	Reserved
2400 0000	2FFF FFFF	0 2400 0000	0 2FFF FFFF	192M	Reserved
3000 0000	331F FFFF	0 3000 0000	0 331F FFFF	50M	Reserved
3320 0000	335F FFFF	0 3320 0000	0 335F FFFF	4M	RAC_A data
3360 0000	33FF FFFF	0 3360 0000	0 33FF FFFF	10M	Reserved
3400 0000	341F FFFF	0 3400 0000	0 341F FFFF	2M	Queue Manager subsystem data
3420 0000	342F FFFF	0 3420 0000	0 342F FFFF	1M	Reserved
3430 0000	3439 FFFF	0 3430 0000	0 3439 FFFF	640K	Reserved

Table 2-2 Memory Map Summary (Part 9 of 9)

Logical 32 bit Address		Physical 36 bit Address		Bytes	Description
Start	End	Start	End		
343A 0000	343F FFFF	0 343A 0000	0 343F FFFF	384K	Reserved
3440 0000	347F FFFF	0 3440 0000	0 347F FFFF	4M	Reserved
3480 0000	34BF FFFF	0 3480 0000	0 34BF FFFF	4M	Reserved
34C0 0000	34C2 FFFF	0 34C0 0000	0 34C2 FFFF	192K	TAC data
34C3 0000	34FF FFFF	0 34C3 0000	0 34FF FFFF	4M-192K	Reserved
3500 0000	3500 03FF	0 3500 0000	0 3500 03FF	1K	Memory protection unit (MPU) 5
3500 0400	3500 7FFF	0 3500 0400	0 3500 7FFF	31K	Reserved
3500 8000	3500 81FF	0 3500 8000	0 3500 81FF	512	Reserved
3500 8200	3501 FFFF	0 3500 8200	0 3501 FFFF	95K	Reserved
3502 0000	3502 03FF	0 3502 0000	0 3502 03FF	1K	TCP3d_C config
3502 0400	3503 FFFF	0 3502 0400	0 3503 FFFF	127K	Reserved
3504 0000	3504 07FF	0 3504 0000	0 3504 07FF	2K	FFTC_C config
3504 0800	350F FFFF	0 3504 0800	0 350F FFFF	766K	Reserved
3510 0000	351F FFFF	0 3510 0000	0 351F FFFF	1M	Reserved
3520 0000	3521 FFFF	0 3520 0000	0 3521 FFFF	128K	BCP config
3522 0000	355F FFFF	0 3522 0000	0 355F FFFF	3968K	Reserved
3560 0000	356F FFFF	0 3560 0000	0 356F FFFF	1M	TCP3d_C data
3570 0000	37FF FFFF	0 3570 0000	0 37FF FFFF	41M	Reserved
3800 0000	3FFF FFFF	0 3800 0000	0 3FFF FFFF	128M	Reserved
4000 0000	4FFF FFFF	0 4000 0000	0 4FFF FFFF	256M	HyperLink data
5000 0000	5FFF FFFF	0 5000 0000	0 5FFF FFFF	256M	Reserved
6000 0000	6FFF FFFF	0 6000 0000	0 6FFF FFFF	256M	PCIe data
7000 0000	73FF FFFF	0 7000 0000	0 73FF FFFF	64M	Reserved
7400 0000	77FF FFFF	0 7400 0000	0 77FF FFFF	64M	Reserved
7800 0000	7BFF FFFF	0 7800 0000	0 7BFF FFFF	64M	Reserved
7C00 0000	7FFF FFFF	0 7C00 0000	0 7FFF FFFF	64M	Reserved
8000 0000	FFFF FFFF	8 0000 0000	8 7FFF FFFF	2G	DDR3 EMIF data
End of Table 2-2					

2.3 Boot Sequence

The boot sequence is a process by which the DSP's internal memory is loaded with program and data sections. The DSP's internal registers are programmed with predetermined values. The boot sequence is started automatically after each power-on reset. A hard reset, soft reset or local reset to an individual C66x CorePac should not affect the state of the hardware boot controller on the device. For more details on the initiators of the resets, see section 7.4 "Reset Controller" on page 122. The bootloader uses a section of the L2 SRAM (start address 0x008F 2DC0 and end address 0x008F FFFF) during initial booting of the device. For more details on the type of configurations stored in this reserved L2 section see the *Bootloader for the C66x DSP User Guide* in "Related Documentation from Texas Instruments" on page 66.

The C6670 supports several boot processes that begins execution at the ROM base address, which contains the bootloader code necessary to support various device boot modes. The boot processes are software-driven and use the BOOTMODE[12:0] device configuration inputs to determine the software configuration that must be completed. For more details on boot sequence see the *Bootloader for the C66x DSP User Guide* in "Related Documentation from Texas Instruments" on page 66.

2.4 Boot Modes Supported and PLL Settings

The device supports several boot processes, which leverage the internal boot ROM. Most boot processes are software driven, using the BOOTMODE[3:0] device configuration inputs to determine the software configuration that must be completed. From a hardware perspective, there are two possible boot modes:

- **Public ROM Boot** - C66x CorePac is released from reset and begins executing from the L3 ROM base address. After performing the boot process (e.g., from I^2C ROM, Ethernet, or RapidIO), the C66x CorePac then begins execution from the L2 RAM base address.
- **Secure ROM Boot** - On secure devices, the C66x CorePac is released from reset and begins executing from secure ROM. Software in the secure ROM will free up internal RAM pages, after which the C66x CorePac initiates the boot process. The C66x CorePac performs any authentication and decryption required on the bootloaded image prior to beginning execution.

The boot process performed by the C66x CorePac in public ROM boot and secure ROM boot is determined by the BOOTMODE[12:0] value in the DEVSTAT register. The C66x CorePac reads this value, and then executes the associated boot process in software. Figure 2-2 shows the bits associated with BOOTMODE[12:0]. PLL settings are shown at the end of this section, and the PLL setup details can be found in Section 7.5 "Main PLL and the PLL Controller" on page 128

Figure 2-2 Boot Mode Pin Decoding

Boot Mode Pins												
12	11	10	9	8	7	6	5	4	3	2	1	0
PLL Mult I^2C /SPI Ext Dev Cfg			Device Configuration					Reserved (1)		Boot Device		

1 BOOTMODE[4:3] are reserved in all modes except No-Boot, Ethernet (SGMII), I^2C and SPI boot mode

2.4.1 Boot Device Field

The Boot Device field BOOTMODE[2:0] defines the boot device that is chosen. Table 2-3 shows the supported boot modes.

Table 2-3 Boot Mode Pins: Boot Device Values

Bit	Field	Description
2-0	Boot Device	Device boot mode 0 = No boot 1 = Serial Rapid I/O 2 = Ethernet (SGMII) (PA driven from core clk) 3 = Ethernet (SGMII) (PA driver from PA clk) 4 = PCI 5 = I^2C 6 = SPI 7 = HyperLink
End of Table 2-3		

2.4.2 Device Configuration Field

The device configuration fields BOOTMODE[9:3] are used to configure the boot peripheral and, therefore, the bit definitions depend on the boot mode.

2.4.2.1 No Boot Device Configuration

Figure 2-3 No Boot Configuration Fields

9	8	7	6	5	4	3
Sub-Mode		Reserved				

Table 2-4 No Boot Configuration Field Descriptions

Bit	Field	Description
9-8	Sub-Mode	Sub mode select 0 = No boot 1 - 3 = Reserved
7-3	Reserved	Reserved
End of Table 2-4		

2.4.2.2 Serial Rapid I/O Boot Device Configuration

The device ID is always set to 0xff (8-bit node IDs) or 0xffff (16 bit node IDs) at power-on reset.

Figure 2-4 Serial Rapid I/O Device Configuration Fields

9	8	7	6	5	4	3
Lane Setup	Data Rate		Ref Clock		Reserved	

Table 2-5 Serial Rapid I/O Configuration Field Descriptions

Bit	Field	Description
9	Lane Setup	SRIO port and lane configuration 0 = Port configured as 4 ports each 1 lane wide (4 -1× ports) 1 = Port configured as 2 ports 2 lanes wide (2 – 2× ports)
8-7	Data Rate	SRIO data rate configuration 0 = 1.25 GBs 1 = 2.5 GBs 2 = 3.125 GBs 3 = 5.0 GBs
6-5	Ref Clock	SRIO reference clock configuration 0 = Reference clock = 156.25 MHz 1 = Reference clock = 250 MHz 2 = Reference clock = 312.5 MHz 3 = Reserved
4-3	Reserved	Reserved
End of Table 2-5		

In SRIO boot mode, both the message mode and DirectIO mode will be enabled by default. If use of the memory reserved for received messages is required and reception of messages cannot be prevented, the master can disable the message mode by writing to the boot table and generating a boot restart.

2.4.2.3 Ethernet (SGMII) Boot Device Configuration

Figure 2-5 Ethernet (SGMII) Device Configuration Fields

9	8	7	6	5	4	3
SerDes Clock Mult		Ext connection		Dev ID		

Table 2-6 Ethernet (SGMII) Configuration Field Descriptions

Bit	Field	Description
9-8	SerDes clock mult	SGMII SerDes input clock. The output frequency of the PLL must be 1.25 GBs. 0 = ×8 for input clock of 156.25 MHz 1 = ×5 for input clock of 250 MHz 2 = ×4 for input clock of 312.5 MHz 3 = Reserved
7-6	Ext connection	External connection mode 0 = MAC to MAC connection, master with auto negotiation 1 = MAC to MAC connection, slave, and MAC to PHY 2 = MAC to MAC, forced link 3 = MAC to fiber connection
5-3	Device ID	This value can range from 0 to 7 and is used in the device ID field of the Ethernet-ready frame.
End of Table 2-6		

2.4.2.4 PCI Boot Device Configuration

Additional device configuration is provided in the PCI bits in the DEVSTAT register.

Figure 2-6 PCI Device Configuration Fields

9	8	7	6	5	4	3
Reserved	BAR Config				Reserved	

Table 2-7 PCI Device Configuration Field Descriptions

Bit	Field	Description
9	Reserved	Reserved
8-5	Bar Config	PCIe BAR registers configuration This value can range from 0 to 0xf. See Table 2-8.
4-3	Reserved	Reserved
End of Table 2-7		

Table 2-8 BAR Config / PCIe Window Sizes

BAR cfg	BAR0	32-Bit Address Translation					64-Bit Address Translation	
		BAR1	BAR2	BAR3	BAR4	BAR5	BAR2/3	BAR4/5
0b0000	PCIe MMRs	32	32	32	32	Clone of BAR4		
0b0001		16	16	32	64			
0b0010		16	32	32	64			
0b0011		32	32	32	64			
0b0100		16	16	64	64			
0b0101		16	32	64	64			
0b0110		32	32	64	64			
0b0111		32	32	64	128			
0b1000		64	64	128	256			
0b1001		4	128	128	128			
0b1010		4	128	128	256			
0b1011		4	128	256	256			
0b1100							256	256
0b1101							512	512
0b1110							1024	1024
0b1111							2048	2048

2.4.2.5 I²C Boot Device Configuration

2.4.2.5.1 I²C Master Mode

In master mode, the I²C device configuration uses ten bits of device configuration instead of seven as used in other boot modes. In this mode, the device will make the initial read of the I²C EEPROM while the PLL is in bypass mode. The initial read will contain the desired clock multiplier, which will be set up prior to any subsequent reads.

Figure 2-7 I²C Master Mode Device Configuration Fields

12	11	10	9	8	7	6	5	4	3
Reserved	Speed	Address	Reserved	Mode	Parameter Index				

Table 2-9 I²C Master Mode Device Configuration Field Descriptions

Bit	Field	Description
12	Reserved	Reserved
11	Speed	I²C data rate configuration 0 = I²C data rate set to approximately 20 kHz 1 = I²C fast mode. Data rate set to approximately 400 kHz (will not exceed)
10	Address	I²C bus address configuration 0 = Boot from I²C EEPROM at I²C bus address 0x50 1 = Boot from I²C EEPROM at I²C bus address 0x51
9	Reserved	Reserved
8	Mode	I²C operation mode 0 = Master mode 1 = Passive mode (see 2.4.2.5.2 "I2C Passive Mode")
7-3	Parameter Index	Identifies the index of the configuration table initially read from the I²C EEPROM. This value can range from 0 to 32.
End of Table 2-9		

2.4.2.5.2 I²C Passive Mode

In passive mode, the device does not drive the clock, but simply acks data received on the specified address.

Figure 2-8 I²C Passive Mode Device Configuration Fields

9	8	7	6	5	4	3
Reserved	Mode	Receive I²C Address			Reserved	

Table 2-10 I²C Passive Mode Device Configuration Field Descriptions

Bit	Field	Description
9	Reserved	Reserved
8	Mode	I² C operation mode 0 = Master mode (see "I2C Master Mode" on page 33) 1 = Passive mode
8-5	Receive I²C Address	I²C bus address configuration 0 - 7 = The I²C bus address the device will listen to for data The actual value on the bus is 0x19 plus the value in bits [8:5]. For Ex. if bits[8:5] = 0 then the device will listen to I²C bus address 0x19.
4-3	Reserved	Reserved
End of Table 2-10		

2.4.2.6 SPI Boot Device Configuration

In SPI boot mode, the SPI device configuration uses ten bits of device configuration instead of seven as used in other boot modes.

Figure 2-9 SPI Device Configuration Fields

12	11	10	9	8	7	6	5	4	3
Mode		4, 5 Pin	Addr Width	Chip Select		Parameter Table Index			

Table 2-11 SPI Device Configuration Field Descriptions

Bit	Field	Description
12-11	Mode	Clk Pol / Phase 0 = Data is output on the rising edge of SPICLK. Input data is latched on the falling edge. 1 = Data is output one half-cycle before the first rising edge of SPICLK and on subsequent falling edges. Input data is latched on the rising edge of SPICLK. 2 = Data is output on the falling edge of SPICLK. Input data is latched on the rising edge. 3 = Data is output one half-cycle before the first falling edge of SPICLK and on subsequent rising edges. Input data is latched on the falling edge of SPICLK.
10	4, 5 Pin	SPI operation mode configuration 0 = 4-pin mode used 1 = 5-pin mode used
9	Addr Width	SPI address width configuration 0 = 16-bit address values are used 1 = 24-bit address values are used
8-7	Chip Select	The chip select field value
6-3	Parameter Table Index	Specifies which parameter table is loaded
End of Table 2-11		

2.4.2.7 HyperLink Boot Device Configuration

Figure 2-10 HyperLink Boot Device Configuration Fields

9	8	7	6	5	4	3
Reserved	Data Rate		Ref Clock		Reserved	

Table 2-12 HyperLink Boot Device Configuration Field Descriptions

Bit	Field	Description
9	Reserved	Reserved
8-7	Data Rate	HyperLink data rate configuration 0 = 1.25 GBs 1 = 3.125 GBs 2 = 6.25 GBs 3 = Reserved
6-5	Ref Clocks	HyperLink reference clock configuration 0 = 156.25 MHz 1 = 250 MHz 2 = 312.5 MHz 3 = Reserved
4-3	Reserved	Reserved
End of Table 2-12		

2.4.3 PLL Settings

The PLL default settings are determined by the BOOTMODE[12:10] bits. Table 2-13 shows settings for various input clock frequencies. This will set the PLL to the maximum clock setting for the device.

$$CLK = CLKIN \times (PLLM+1) \div (2 \times (PLLD+1))$$

The configuration for the PASS PLL is also shown. The PASS PLL is configured with these values only if the Ethernet boot mode is selected with the input clock set to match the main PLL clock (not the SGMII SerDes clock). See Table 2-3 for details on configuring Ethernet boot mode. The output from the PASS PLL goes through an on-chip divider to reduce the operating frequency before reaching the NETCP. The PASS PLL generates 1050 MHz, and after the chip divider (/3), applies 350 MHz to the NETCP.

The Main PLL is controlled using a PLL controller and a chip-level MMR. The DDR3 PLL and PASS PLL are controlled by chip level MMRs. For details on how to set up the PLL see Section 7.5 "Main PLL and the PLL Controller" on page 128. For details on the operation of the PLL controller module, see the *Phase Locked Loop (PLL) Controller for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

Table 2-13 C66x CorePac System PLL Configuration

BOOTMODE [12:10]	Input Clock Freq (MHz)	800 MHz Device			1000 MHz Device			1200 MHz Device			PA = 350 MHz (1)		
		PLLD	PLLM	DSP ƒ	PLLD	PLLM	DSP ƒ	PLLD	PLLM	DSP ƒ	PLLD	PLLM	DSP ƒ (2)
0b000	50.00	0	31	800	0	39	1000	0	47	1200	0	41	1050
0b001	66.67	0	23	800.04	0	29	1000.05	0	35	1200.06	1	62	1050.053
0b010	80.00	0	19	800	0	24	1000	0	29	1200	3	104	1050
0b011	100.00	0	15	800	0	19	1000	0	23	1200	0	20	1050
0b100	156.25	24	255	800	4	63	1000	24	383	1200	24	335	1050
0b101	250.00	4	31	800	0	7	1000	4	47	1200	4	41	1050
0b110	312.50	24	127	800	4	31	1000	24	191	1200	24	167	1050
0b111	122.88	47	624	800	28	471	999.989	31	624	1200	11	204	1049.6

1 The PASS PLL generates 1050 MHz and is internally divided by 3 to feed 350 MHz to the packet accelerator.
2 ƒ represents frequency in MHz.

2.5 Second-Level Bootloaders

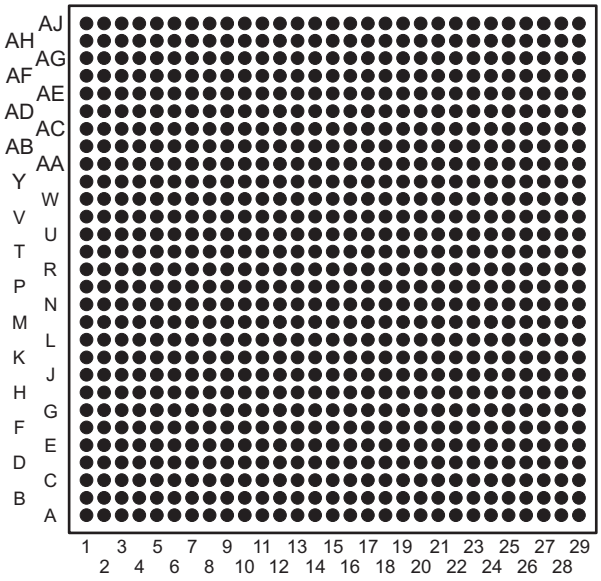
Any of the boot modes can be used to download a second-level bootloader. A second-level bootloader allows for any level of customization to current boot methods as well as the definition of a completely customized boot.

2.6 Terminals

2.6.1 Package Terminals

Figure 2-11 shows the TMS320C6670 CYP ball grid array package (bottom view).

Figure 2-11 CYP 841-PIN BGA Package (Bottom View)



2.6.2 Pin Map

Figure 2-13 through Figure 2-16 show the TMS320C6670 pin assignments in four quadrants (A, B, C, and D).

Figure 2-12 Pin Map Quadrants (Bottom View)

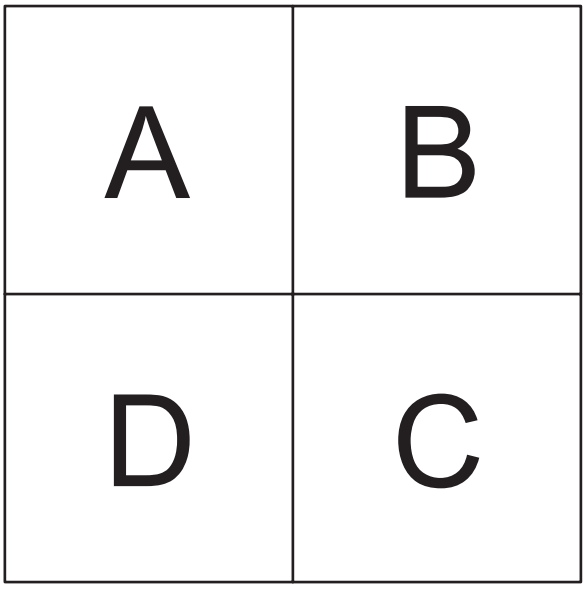


Figure 2-13 Upper Left Quadrant—A (Bottom View)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
AJ	VSS	CVDD1	VSS	SGMII1 RXN	SGMII1RXP	VSS	RIORXN2	RIORXP2	VSS	RIORXP0	RIORXN0	VSS	PCIERXP0	PCIERXN0	VSS
AH	CVDD1	VSS	SGMII0 RXN	SGMII0 RXP	VSS	RIORXN3	RIORXP3	VSS	RIORXP1	RIORXN1	VSS	PCIERXN1	PCIERXP1	VSS	CORESEL0
AG	VSS	CVDD1	VSS	SGMII1 TXN	SGMII1TXP	VSS	RIOTXN2	RIOTXP2	VSS	RIOTXP0	RIOTXN0	VSS	PCIETXP0	PCIETXN0	VSS
AF	CVDD1	VSS	SGMII0 TXN	SGMII0 TXP	VSS	RIOTXN3	RIOTXP3	VSS	RIOTXN1	RIOTXP1	VSS	PCIETXN1	PCIETXP1	VSS	VSS
AE	VSS	CVDD1	VSS	VDDT2	RSV17	VDDR3	VSS	RSV15	VSS	VDDT2	VDDR4	VDDT2	VSS	RSV16	VDDR2
AD	CVDD1	VSS	CVDD1	VSS	VDDT2	VSS	VDDT2	VSS	VDDT2	VSS	VDDT2	VSS	VDDT2	VSS	CORESEL2
AC	VSS	DVDD18	VSS	DVDD18	VSS	VDDT2	VSS	VDDT2	VSS	VDDT2	VSS	VDDT2	VSS	VDDT2	VSS
AB	VCNTL3	VSS	VCNTL1	VCNTL0	DVDD18	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	AVDDA3
AA	MCMTX PMDAT	MCMTX PMCLK	MCMRX PMCLK	VCNTL2	VSS	VSS	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS
Y	MCMTX FLCLK	MCMTX FLDAT	MCMRX PMDAT	RSV28	DVDD18	VSS	VDDT1	VSS	CVDD	VSS	CVDD1	VSS	CVDD1	VSS	CVDD1
W	MCMCLKP	MCMCLKN	MCMRX FLDAT	RSV29	VSS	VDDT1	VSS	CVDD	VSS	CVDD	VSS	CVDD1	VSS	CVDD1	VSS
V	MCMREF CLKOUTN	MCMREF CLKOUTP	MCMRX FLCLK	RSV14	DVDD18	VSS	VDDT1	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD
U	VSS	VSS	VSS	VSS	VSS	VSS	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS
T	VSS	MCMRXN0	VSS	VSS	MCMTXN0	VSS	VDDT1	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD

A	

Figure 2-14 Upper Right Quadrant—B (Bottom View)

16	17	18	19	20	21	22	23	24	25	26	27	28	29	
SRIOSGMII CLKP	PCIECLKN	PASSCLKP	GPIO08	GPIO00	SPIDOUT	SPISCS1	TIMI0	UARTTXD	RSV01	EMU17	EMU15	DVDD18	VSS	AJ
SRIOSGMII CLKN	PCIECLKP	PASSCLKN	GPIO07	GPIO14	SPISCS0	SPIDIN	TIMO0	UARTCTS	EMU18	DVDD18	EMU16	EMU14	DVDD18	AH
MDIO	RSV24	GPIO01	GPIO10	GPIO06	SPICLK	VSS	TIMI1	UARTRTS	EMU13	VSS	EMU12	EMU11	EMU09	AG
MDCLK	RSV25	GPIO04	DVDD18	VSS	GPIO13	DVDD18	TIMO1	UARTRXD	EMU06	EMU10	EMU08	EMU03	EMU01	AF
RSV22	EXTFRAME EVENT	GPIO05	GPIO03	GPIO12	GPIO09	$\overline{\text{LRESET}}$	$\overline{\text{RESETFULL}}$	DVDD18	EMU07	EMU04	DVDD18	EMU02	EMU00	AE
RSV23	SDA	$\overline{\text{RESETSTAT}}$	GPIO02	GPIO11	GPIO15	RSV12	PACLKSEL	VSS	EMU05	$\overline{\text{TRST}}$	VSS	TDI	TCK	AD
CORESEL1	SCL	HOUT	$\overline{\text{POR}}$	$\overline{\text{LRESET}}$ NMIEN	BOOT COMPLETE	RSV13	RSV03	$\overline{\text{RESET}}$	$\overline{\text{NMI}}$	TMS	TDO	SYSCLKN	SYSCLKP	AC
VSS	DVDD18	VSS	DVDD18	VSS	DVDD18	VSS	CVDD	VSS	CORE CLKSEL	RSV20	PHYSYNC	ALTCORE CLKN	ALTCORE CLKP	AB
CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	RSV18	VSS	SYSCLKOUT	RADSYNC	RP1CLKN	RP1FBN	AA
VSS	CVDD1	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	DVDD18	VSS	RSV04	RP1CLKP	RP1FBP	Y
CVDD1	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	AVDDA1	VSS	VSS	RSV05	VSS	VSS	W
VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	VDDT3	AIFTXP5	VSS	AIFRXP5	VSS	V
CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	VDDT3	RSV26	AIFTXN5	AIFTXN4	AIFRXN5	AIFRXP4	U
VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	VDDT3	VSS	VDDT3	VSS	AIFTXP4	VSS	AIFRXN4	T

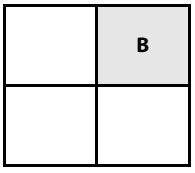
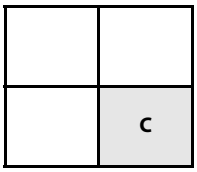
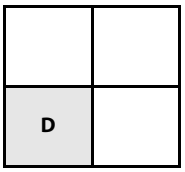


Figure 2-15 Lower Right Quadrant—C (Bottom View)



CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	VDDT3	VDDR5	AIFTXN2	VSS	AIFRXN2	VSS	R
VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	VDDT3	VSS	VDDT3	AIFTXP2	AIFTXN3	AIFRXP2	AIFRXN3	P
CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	VDDT3	VDDR6	VSS	AIFTXP3	VSS	AIFRXP3	N
VSS	CVDD	VSS	CVDD	VSS	CVDD1	VSS	VDDT3	VSS	VDDT3	AIFTXP0	VSS	AIFRXP0	VSS	M
CVDD1	VSS	CVDD	VSS	CVDD1	VSS	CVDD1	VSS	VDDT3	RSV27	AIFTXN0	AIFTXN1	AIFRXN0	AIFRXP1	L
VSS	CVDD1	VSS	CVDD	VSS	CVDD1	VSS	RSV0B	RSV0A	VDDT3	VSS	AIFTXP1	VSS	AIFRXN1	K
CVDD1	VSS	CVDD	VSS	CVDD1	VSS	CVDD1	RSV11	RSV08	RSV09	AVDDA2	VSS	RSV06	VSS	J
VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	RSV10	PTV15	DVDD18	DDRSL RATE1	DDRSL RATE0	RSV07	DDRCLKN	H
DVDD15	VSS	DVDD15	VSS	DVDD15	VSS	DVDD15	VSS	DVDD15	VSS	RSV21	RSV19	DVDD18	DDRCLKP	G
VSS	DVDD15	VSS	DVDD15	DDRD25	DDRD27	DDRD17	DDRD16	DDRD08	DDRD07	VSS	DVDD15	VSS	VSS	F
DDRA10	DDRA12	DDRCKE1	DDRCB00	VSS	DDRD26	DDRD23	DDRD19	DDRD09	DDRD10	DDRD06	DDRD02	DDRD00	DDRDQM0	E
DDRA11	DDRA14	VSS	DDRCB02	DVDD15	DDRD24	DDRD28	DVDD15	DDRD18	DDRD11	DDRD12	DDRD04	DDRD03	DDRD01	D
DDRA13	DDRA15	DDRCB05	DDRCB04	DDRCB01	DDRD29	DDRD31	VSS	DDRD22	DVDD15	DDRD13	DDRDQM1	DDRDQS0P	DDRDQS0N	C
DDRCLK OUTN1	VSS	DDRCB06	DDRDQS8N	DDRCB03	DDRDQS3N	DDRD30	DDRD21	DDRDQS2N	VSS	DDRD14	DDRDQS1N	DDRD05	DVDD15	B
DDRCLK OUTP1	DVDD15	DDRCB07	DDRDQS8P	DDRDQM8	DDRDQS3P	DDRDQM3	DDRD20	DDRDQS2P	DDRDQM2	DDRD15	DDRDQS1P	DVDD15	VSS	A
16	17	18	19	20	21	22	23	24	25	26	27	28	29	

Figure 2-16 Lower Left Quadrant—D (Bottom View)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	MCMRXP1	MCMRXP0	VSS	MCMTXN1	MCMTXP0	VDDT1	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS
P	MCMRXN1	VSS	VSS	MCMTXP1	VSS	VSS	VDDT1	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD
N	VSS	MCMRXN3	VSS	VSS	MCMTXP3	VDDT1	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS
M	MCMRXP2	MCMRXP3	VSS	MCMTXP2	MCMTXN3	VSS	VDDT1	VSS	CVDD1	VSS	CVDD	VSS	CVDD	VSS	CVDD
L	MCMRXN2	VSS	VSS	MCMTXN2	VSS	VSS	VSS	CVDD1	VSS	CVDD1	VSS	CVDD	VSS	CVDD1	VSS
K	VSS	VSS	VSS	VSS	VSS	VDDR1	CVDD1	VSS	CVDD1	VSS	CVDD	VSS	CVDD1	VSS	CVDD1
J	VSS	CVDD1	VSS	CVDD1	VSS	CVDD1	VSS	CVDD1	VSS	CVDD1	VSS	CVDD	VSS	CVDD1	VSS
H	CVDD1	VSS	CVDD1	VSS	CVDD1	VSS	CVDD1	VSS	CVDD1	VSS	CVDD	VSS	CVDD	VSS	CVDD
G	VSS	DVDD15	VSS	DVDD15	VSS	CVDD1	VSS	DVDD15	VSS	DVDD15	VSS	DVDD15	VSS	DVDD15	VSS
F	DDRD63	DDRD60	DDRD61	DDRD56	DVDD15	VSS	DVDD15	VSS	DVDD15	VSS	DVDD15	VSS	DDRA03	DDRA02	DDRA08
E	DDRD62	DDRD58	DVDD15	DDRD53	VSS	DDRD45	DDRD42	DDRD39	DDRD36	DDRD32	$\overline{\text{DDRRESET}}$	$\overline{\text{DDRWE}}$	DDRODT1	VREFSSTL	DDRA09
D	DDRDQS7P	DDRD57	VSS	DDRD52	DVDD15	DDRD46	DDRD41	DVDD15	DDRD35	DDRD33	DDRCKE0	$\overline{\text{DDRCAS}}$	DDRODT0	VSS	DDRA07
C	DDRDQS7N	DDRD59	DDRD55	DDRD54	DDRD48	DDRD47	DDRD43	VSS	DDRD37	$\overline{\text{DDRRAS}}$	$\overline{\text{DDRCE0}}$	$\overline{\text{DDRCE1}}$	DDRBA2	DVDD15	DDRA05
B	DVDD15	DDRDQM7	DDRDQS6P	DDRD50	DDRDQM6	DDRDQS5P	DDRD44	DDRD38	DDRDQS4N	DDRD34	VSS	DDRCLK OUTN0	DDRBA1	DDRA01	DDRA06
A	VSS	DVDD15	DDRDQS6N	DDRD51	DDRD49	DDRDQS5N	DDRD40	DDRDQM5	DDRDQS4P	DDRDQM4	DVDD15	DDRCLK OUTP0	DDRBA0	DDRA00	DDRA04

2.7 Terminal Functions

The terminal functions table (Table 2-15) identifies the external signal names, the associated pin (ball) numbers, the pin type (I, O/Z, or I/O/Z), whether the pin has any internal pullup/pulldown resistors, and gives functional pin descriptions. This table is arranged by function. The power terminal functions table (Table 2-16) lists the various power supply pins and ground pins and gives functional pin descriptions. Table 2-17 shows all pins arranged by signal name. Table 2-18 shows all pins arranged by ball number.

There are 17 pins that have a secondary function as well as a primary function. The secondary function is indicated with a dagger (†).

For more detailed information on device configuration, peripheral selection, multiplexed/shared pins, and pullup/pulldown resistors, see chapter 3 "Device Configuration" on page 67.

Use the symbol definitions in Table 2-14 when reading Table 2-15.

Table 2-14 I/O Functional Symbol Definitions

Functional Symbol	Definition	Table 2-15 Column Heading
IPD or IPU	Internal 100-µA pulldown or pullup is provided for this terminal. In most systems, a 1-kΩ resistor can be used to oppose the IPD/IPU. For more detailed information on pulldown/pullup resistors and situations in which external pulldown/pullup resistors are required, see the *Hardware Design Guide for KeyStone Devices* in 2.9 "Related Documentation from Texas Instruments" on page 66.	IPD/IPU
A	Analog signal	Type
GND	Ground	Type
I	Input terminal	Type
O	Output terminal	Type
S	Supply voltage	Type
Z	Three-state terminal or high impedance	Type
End of Table 2-14		

Table 2-15 Terminal Functions — Signals and Control by Function (Part 1 of 12)

Signal Name	Ball No.	Type	IPD/IPU	Description
AIF				
AIFRXN0	L28	I		Antenna interface receive data (6 links)
AIFRXP0	M28	I		
AIFRXN1	K29	I		
AIFRXP1	L29	I		
AIFRXN2	R28	I		
AIFRXP2	P28	I		
AIFRXN3	P29	I		
AIFRXP3	N29	I		
AIFRXN4	T29	I		
AIFRXP4	U29	I		
AIFRXN5	U28	I		
AIFRXP5	V28	I		

Table 2-15 Terminal Functions — Signals and Control by Function (Part 2 of 12)

Signal Name	Ball No.	Type	IPD/IPU	Description
AIFTXN0	L26	O		Antenna interface transmit data (6 links)
AIFTXP0	M26	O		
AIFTXN1	L27	O		
AIFTXP1	K27	O		
AIFTXN2	R26	O		
AIFTXP2	P26	O		
AIFTXN3	P27	O		
AIFTXP3	N27	O		
AIFTXN4	U27	O		
AIFTXP4	T27	O		
AIFTXN5	U26	O		
AIFTXP5	V26	O		
AIF2 Timer (AT) Module				
RP1CLKP	Y28	I		Frame sync interface clock used to drive the frame synchronization interface (OBSAI RP1 clock)
RP1CLKN	AA28	I		
EXTFRAMEEVENT	AE17	OZ	Down	Frame sync clock output
RP1FBP	Y29	I		Frame burst to drive frame indicators to the frame synchronization module (OBSAI RP1)
RP1FBN	AA29	I		
PHYSYNC	AB27	I	Down	Frame sync input for PHY timer
RADSYNC	AA27	I	Down	Frame sync input for radio timer
Boot Configuration Pins				
LENDIAN †	AJ20	IOZ	Up	Endian configuration pin (pin shared with GPIO[0])
BOOTMODE00 †	AG18	IOZ	Down	See Section 2.4 "Boot Modes Supported and PLL Settings" on page 30 for more details (Pins shared with GPIO[1:13])
BOOTMODE01†	AD19	IOZ	Down	
BOOTMODE02 †	AE19	IOZ	Down	
BOOTMODE03 †	AF18	IOZ	Down	
BOOTMODE04 †	AE18	IOZ	Down	
BOOTMODE05 †	AG20	IOZ	Down	
BOOTMODE06 †	AH19	IOZ	Down	
BOOTMODE07 †	AJ19	IOZ	Down	
BOOTMODE08 †	AE21	IOZ	Down	
BOOTMODE09 †	AG19	IOZ	Down	
BOOTMODE10 †	AD20	IOZ	Down	
BOOTMODE11 †	AE20	IOZ	Down	
BOOTMODE12 †	AF21	IOZ	Down	
PCIESSMODE0 †	AH20	IOZ	Down	PCIe mode selection pins (pins shared with GPIO[14:15])
PCIESSMODE1 †	AD21	IOZ	Down	
PCIESSEN †	AJ23	I		PCIe module enable (pin shared with TIMI0)
Clock / Reset				
SYSCLKP	AC29	I		System clock input to antenna interface and/or main PLL
SYSCLKN	AC28	I		
PASSCLKP	AJ18	I		Network coprocessor reference clock to PASS PLL
PASSCLKN	AH18	I		

Table 2-15 Terminal Functions — Signals and Control by Function (Part 3 of 12)

Signal Name	Ball No.	Type	IPD/IPU	Description
ALTCORECLKP	AB29	I		Alternate System clock input to main PLL
ALTCORECLKN	AB28	I		
SRIOSGMIICLKP	AJ16	I		RapidIO/SGMII reference clock to drive the RapidIO and SGMII SerDes
SRIOSGMIICLKN	AH16	I		
DDRCLKP	G29	I		DDR reference clock input to DDR PLL
DDRCLKN	H29	I		
PCIECLKP	AH17	I		PCIe reference clock input to drive PCIe SerDes
PCIECLKN	AJ17	I		
MCMCLKP	W1	I		HyperLink reference clock input to drive the HyperLink SerDes
MCMCLKN	W2	I		
SYSCLKOUT	AA26	OZ	Down	System clock output to be used as a general purpose output clock for debug purposes
CORECLKSEL	AB25	I	Down	Core clock select to select between SYSCLK and ALTCORECCLK to the main PLL
PACLKSEL	AD23	IOZ	Down	PA clock select to choose between PASSCLK and the output of main PLL MUX (dependent on CORECLKSEL pin) to the PA sub-system PLL
HOUT	AC18	OZ	Up	Interrupt output pulse created by IPCGRH
$\overline{\text{NMI}}$	AC25	I	Up	Non-maskable Interrupt
$\overline{\text{LRESET}}$	AE22	I	Up	Local Reset
$\overline{\text{LRESETNMIEN}}$	AC20	I	Up	Enable for core selects
CORESEL0	AH15	I	Down	Select for the target core for LRESET and NMI. For more details see Table 7-47 "NMI and LRESET Timing Requirements" on page 178
CORESEL1	AC16	I	Down	
CORESEL2	AD15	I	Down	
$\overline{\text{RESETFULL}}$	AE23	I	Up	Full reset power-on reset
$\overline{\text{RESET}}$	AC24	I	Up	Reset of non isolated portion on the device
$\overline{\text{POR}}$	AC19	I		POR (power-on reset)
$\overline{\text{RESETSTAT}}$	AD18	O	Up	Reset status output
BOOTCOMPLETE	AC21	OZ	Down	Boot progress indication output
PTV15	H24	A		PTV Compensation NMOS Reference Input. A precision resistor placed between the PTV15 pin and ground is used to closely tune the output impedance of the DDR interface drivers to 50 Ohms. Presently, the recommended value for this 1% resistor is 45.3 Ohms.
DDR				
DDRDQM0	E29	OZ		DDR EMIF data masks
DDRDQM1	C27	OZ		
DDRDQM2	A25	OZ		
DDRDQM3	A22	OZ		
DDRDQM4	A10	OZ		
DDRDQM5	A8	OZ		
DDRDQM6	B5	OZ		
DDRDQM7	B2	OZ		
DDRDQM8	A20	OZ		

Table 2-15 Terminal Functions — Signals and Control by Function (Part 4 of 12)

Signal Name	Ball No.	Type	IPD/IPU	Description
DDRDQS0P	C28	IOZ		DDR EMIF data strobe
DDRDQS0N	C29	IOZ		
DDRDQS1P	A27	IOZ		
DDRDQS1N	B27	IOZ		
DDRDQS2P	A24	IOZ		
DDRDQS2N	B24	IOZ		
DDRDQS3P	A21	IOZ		
DDRDQS3N	B21	IOZ		
DDRDQS4P	A9	IOZ		
DDRDQS4N	B9	IOZ		
DDRDQS5P	B6	IOZ		
DDRDQS5N	A6	IOZ		
DDRDQS6P	B3	IOZ		
DDRDQS6N	A3	IOZ		
DDRDQS7P	D1	IOZ		
DDRDQS7N	C1	IOZ		
DDRDQS8P	A19	IOZ		
DDRDQS8N	B19	IOZ		
DDRCB00	E19	IOZ		DDR EMIF check bits
DDRCB01	C20	IOZ		
DDRCB02	D19	IOZ		
DDRCB03	B20	IOZ		
DDRCB04	C19	IOZ		
DDRCB05	C18	IOZ		
DDRCB06	B18	IOZ		
DDRCB07	A18	IOZ		

Table 2-15 Terminal Functions — Signals and Control by Function (Part 5 of 12)

Signal Name	Ball No.	Type	IPD/IPU	Description
DDRD00	E28	IOZ		DDR EMIF data bus
DDRD01	D29	IOZ		
DDRD02	E27	IOZ		
DDRD03	D28	IOZ		
DDRD04	D27	IOZ		
DDRD05	B28	IOZ		
DDRD06	E26	IOZ		
DDRD07	F25	IOZ		
DDRD08	F24	IOZ		
DDRD09	E24	IOZ		
DDRD10	E25	IOZ		
DDRD11	D25	IOZ		
DDRD12	D26	IOZ		
DDRD13	C26	IOZ		
DDRD14	B26	IOZ		
DDRD15	A26	IOZ		
DDRD16	F23	IOZ		
DDRD17	F22	IOZ		
DDRD18	D24	IOZ		
DDRD19	E23	IOZ		
DDRD20	A23	IOZ		
DDRD21	B23	IOZ		
DDRD22	C24	IOZ		
DDRD23	E22	IOZ		
DDRD24	D21	IOZ		
DDRD25	F20	IOZ		
DDRD26	E21	IOZ		
DDRD27	F21	IOZ		
DDRD28	D22	IOZ		
DDRD29	C21	IOZ		

Table 2-15 Terminal Functions — Signals and Control by Function (Part 6 of 12)

Signal Name	Ball No.	Type	IPD/IPU	Description
DDRD30	B22	IOZ		DDR EMIF data bus
DDRD31	C22	IOZ		
DDRD32	E10	IOZ		
DDRD33	D10	IOZ		
DDRD34	B10	IOZ		
DDRD35	D9	IOZ		
DDRD36	E9	IOZ		
DDRD37	C9	IOZ		
DDRD38	B8	IOZ		
DDRD39	E8	IOZ		
DDRD40	A7	IOZ		
DDRD41	D7	IOZ		
DDRD42	E7	IOZ		
DDRD43	C7	IOZ		
DDRD44	B7	IOZ		
DDRD45	E6	IOZ		
DDRD46	D6	IOZ		
DDRD47	C6	IOZ		
DDRD48	C5	IOZ		
DDRD49	A5	IOZ		
DDRD50	B4	IOZ		
DDRD51	A4	IOZ		
DDRD52	D4	IOZ		
DDRD53	E4	IOZ		
DDRD54	C4	IOZ		
DDRD55	C3	IOZ		
DDRD56	F4	IOZ		
DDRD57	D2	IOZ		
DDRD58	E2	IOZ		
DDRD59	C2	IOZ		
DDRD60	F2	IOZ		
DDRD61	F3	IOZ		
DDRD62	E1	IOZ		
DDRD63	F1	IOZ		
$\overline{\text{DDRCE0}}$	C11	OZ		DDR EMIF chip enables
$\overline{\text{DDRCE1}}$	C12	OZ		
DDRBA0	A13	OZ		DDR EMIF bank address
DDRBA1	B13	OZ		
DDRBA2	C13	OZ		

Table 2-15 Terminal Functions — Signals and Control by Function (Part 7 of 12)

Signal Name	Ball No.	Type	IPD/IPU	Description
DDRA00	A14	OZ		DDR EMIF address bus
DDRA01	B14	OZ		
DDRA02	F14	OZ		
DDRA03	F13	OZ		
DDRA04	A15	OZ		
DDRA05	C15	OZ		
DDRA06	B15	OZ		
DDRA07	D15	OZ		
DDRA08	F15	OZ		
DDRA09	E15	OZ		
DDRA10	E16	OZ		
DDRA11	D16	OZ		
DDRA12	E17	OZ		
DDRA13	C16	OZ		
DDRA14	D17	OZ		
DDRA15	C17	OZ		
$\overline{\text{DDRCAS}}$	D12	OZ		DDR EMIF column address strobe
$\overline{\text{DDRRAS}}$	C10	OZ		DDR EMIF row address strobe
$\overline{\text{DDRWE}}$	E12	OZ		DDR EMIF write enable
DDRCKE0	D11	OZ		DDR EMIF clock enables
DDRCKE1	E18	OZ		
DDRCLKOUTP0	A12	OZ		DDR EMIF output clocks to drive SDRAMs (one clock pair per SDRAM)
DDRCLKOUTN0	B12	OZ		
DDRCLKOUTP1	A16	OZ		
DDRCLKOUTN1	B16	OZ		
DDRODT0	D13	OZ		DDR EMIF on-die termination outputs used to set termination on the SDRAMs
DDRODT1	E13	OZ		
$\overline{\text{DDRRESET}}$	E11	OZ		DDR reset signal
DDRSLRATE0	H27	I	Down	DDR slew rate control
DDRSLRATE1	H26	I	Down	
VREFSSTL	E14	P		Reference voltage input for SSTL15 buffers used by DDR EMIF (VDDS15 ÷ 2)

Table 2-15 Terminal Functions — Signals and Control by Function (Part 8 of 12)

Signal Name	Ball No.	Type	IPD/IPU	Description
EMU				
EMU00	AE29	IOZ	Up	Emulation and trace ports
EMU01	AF29	IOZ	Up	
EMU02	AE28	IOZ	Up	
EMU03	AF28	IOZ	Up	
EMU04	AE26	IOZ	Up	
EMU05	AD25	IOZ	Up	
EMU06	AF25	IOZ	Up	
EMU07	AE25	IOZ	Up	
EMU08	AF27	IOZ	Up	
EMU09	AG29	IOZ	Up	
EMU10	AF26	IOZ	Up	
EMU11	AG28	IOZ	Up	
EMU12	AG27	IOZ	Up	
EMU13	AG25	IOZ	Up	
EMU14	AH28	IOZ	Up	
EMU15	AJ27	IOZ	Up	
EMU16	AH27	IOZ	Up	
EMU17	AJ26	IOZ	Up	
EMU18	AH25	IOZ	Up	
General Purpose Input/Output (GPIO)				
GPIO00	AJ20	IOZ	Up	General purpose input/output These GPIO pins have secondary functions assigned to them as mentioned in the ***Boot Configuration Pins*** section above.
GPIO01	AG18	IOZ	Down	
GPIO02	AD19	IOZ	Down	
GPIO03	AE19	IOZ	Down	
GPIO04	AF18	IOZ	Down	
GPIO05	AE18	IOZ	Down	
GPIO06	AG20	IOZ	Down	
GPIO07	AH19	IOZ	Down	
GPIO08	AJ19	IOZ	Down	
GPIO09	AE21	IOZ	Down	
GPIO10	AG19	IOZ	Down	
GPIO11	AD20	IOZ	Down	
GPIO12	AE20	IOZ	Down	
GPIO13	AF21	IOZ	Down	
GPIO14	AH20	IOZ	Down	
GPIO15	AD21	IOZ	Down	

Table 2-15 Terminal Functions — Signals and Control by Function (Part 9 of 12)

Signal Name	Ball No.	Type	IPD/IPU	Description
HyperLink				
MCMRXN0	T2	I		Serial HyperLink receive data (4 links)
MCMRXP0	R2	I		
MCMRXN1	P1	I		
MCMRXP1	R1	I		
MCMRXN2	L1	I		
MCMRXP2	M1	I		
MCMRXN3	N2	I		
MCMRXP3	M2	I		
MCMTXN0	T5	O		Serial HyperLink transmit data (4 links)
MCMTXP0	R5	O		
MCMTXN1	R4	O		
MCMTXP1	P4	O		
MCMTXN2	L4	O		
MCMTXP2	M4	O		
MCMTXN3	M5	O		
MCMTXP3	N5	O		
MCMRXFLCLK	V3	O	Down	Serial HyperLink sideband signals
MCMRXFLDAT	W3	O	Down	
MCMTXFLCLK	Y1	I	Down	
MCMTXFLDAT	Y2	I	Down	
MCMRXPMCLK	AA3	I	Down	
MCMRXPMDAT	Y3	I	Down	
MCMTXPMCLK	AA2	O	Down	
MCMTXPMDAT	AA1	O	Down	
MCMREFCLKOUTP	V2	O		HyperLink reference clock output for daisy chain connection
MCMREFCLKOUTN	V1	O		
I^2C				
SCL	AC17	IOZ		I^2C clock
SDA	AD17	IOZ		I^2C data
JTAG				
TCK	AD29	I	Up	JTAG clock input
TDI	AD28	I	Up	JTAG data input
TDO	AC27	OZ	Up	JTAG data output
TMS	AC26	I	Up	JTAG test mode input
$\overline{TRST}$	AD26	I	Down	JTAG reset
MDIO				
MDIO	AG16	IOZ	Up	MDIO data
MDCLK	AF16	O	Down	MDIO clock

Table 2-15 Terminal Functions — Signals and Control by Function (Part 10 of 12)

Signal Name	Ball No.	Type	IPD/IPU	Description
PCIe				
PCIERXN0	AJ14	I		PCIexpress receive data (2 links)
PCIERXP0	AJ13	I		
PCIERXN1	AH12	I		
PCIERXP1	AH13	I		
PCIETXN0	AG14	O		PCIexpress transmit data (2 links)
PCIETXP0	AG13	O		
PCIETXN1	AF12	O		
PCIETXP1	AF13	O		
Serial RapidIO				
RIORXN0	AJ11	I		Serial RapidIO receive data (4 links)
RIORXP0	AJ10	I		
RIORXN1	AH10	I		
RIORXP1	AH9	I		
RIORXN2	AJ7	I		
RIORXP2	AJ8	I		
RIORXN3	AH6	I		
RIORXP3	AH7	I		
RIOTXN0	AG11	O		Serial RapidIO transmit data (4 links)
RIOTXP0	AG10	O		
RIOTXN1	AF9	O		
RIOTXP1	AF10	O		
RIOTXN2	AG7	O		
RIOTXP2	AG8	O		
RIOTXN3	AF6	O		
RIOTXP3	AF7	O		
SGMII				
SGMII0RXN	AH3	I		Ethernet MAC SGMII receive data (2 links)
SGMII0RXP	AH4	I		
SGMII1RXN	AJ4	I		
SGMII1RXP	AJ5	I		
SGMII0TXN	AF3	O		Ethernet MAC SGMII transmit data (2 links)
SGMII0TXP	AF4	O		
SGMII1TXN	AG4	O		
SGMII1TXP	AG5	O		
SmartReflex				
VCNTL0	AB4	OZ		Voltage control outputs to variable core power supply
VCNTL1	AB3	OZ		
VCNTL2	AA4	OZ		
VCNTL3	AB1	OZ		

Table 2-15 Terminal Functions — Signals and Control by Function (Part 11 of 12)

Signal Name	Ball No.	Type	IPD/IPU	Description
SPI				
SPISCS0	AH21	OZ	Up	SPI interface enable 0
SPISCS1	AJ22	OZ	Up	SPI interface enable 1
SPICLK	AG21	OZ	Down	SPI clock
SPIDIN	AH22	I	Down	SPI data in
SPIDOUT	AJ21	OZ	Down	SPI data out
Timer				
TIMI0	AJ23	I	Down	Timer inputs
TIMI1	AG23	I	Down	
TIMO0	AH23	OZ	Down	Timer outputs
TIMO1	AF23	OZ	Down	
UART				
UARTRXD	AF24	I	Down	UART serial data in
UARTTXD	AJ24	OZ	Down	UART serial data out
UARTCTS	AH24	I	Down	UART clear to send
UARTRTS	AG24	OZ	Down	UART request to send
Reserved				
RSV0A	K24			Reserved - leave unconnected
RSV0B	K23			Reserved - leave unconnected
RSV01	AJ25	IOZ	Down	Reserved - pullup to DVDD18
RSV03	AC23	OZ	Down	Reserved - leave unconnected
RSV04	Y27	O		Reserved - leave unconnected
RSV05	W27	O		Reserved - leave unconnected
RSV06	J28	O		Reserved - leave unconnected
RSV07	H28	O		Reserved - leave unconnected
RSV08	J24	A		Reserved - connect to GND
RSV09	J25	A		Reserved - leave unconnected
RSV10	H23	A		Reserved - leave unconnected
RSV11	J23	A		Reserved - leave unconnected
RSV12	AD22	A		Reserved - leave unconnected
RSV13	AC22	A		Reserved - leave unconnected
RSV14	V4	A		Reserved - leave unconnected
RSV15	AE8	A		Reserved - leave unconnected
RSV16	AE14	A		Reserved - leave unconnected
RSV17	AE5	A		Reserved - leave unconnected
RSV18	AA24	A		Reserved - leave unconnected
RSV19	G27	A		Reserved - leave unconnected
RSV20	AB26	OZ	Down	Reserved - leave unconnected
RSV21	G26	OZ	Down	Reserved - leave unconnected
RSV22	AE16	OZ	Down	Reserved - leave unconnected
RSV23	AD16	A		Reserved - leave unconnected
RSV24	AG17	O		Reserved - leave unconnected
RSV25	AF17	O		Reserved - leave unconnected
RSV26	U25	A		Reserved - leave unconnected

Table 2-15 Terminal Functions — Signals and Control by Function (Part 12 of 12)

Signal Name	Ball No.	Type	IPD/IPU	Description
RSV27	L25	A		Reserved - leave unconnected
RSV28	Y4	IOZ		Reserved - leave unconnected
RSV29	W4	IOZ		Reserved - leave unconnected
End of Table 2-15				

Table 2-16 Terminal Functions — Power and Ground

Supply	Ball No.	Volts	Description
AVDDA1	W24	1.8	PLL supply: Main PLL
AVDDA2	J26	1.8	PLL supply: DDR3 PLL
AVDDA3	AB15	1.8	PLL supply: PASS PLL
CVDD	H11, H13, H15, H17, H19, H21, J12, J18, K11, K19, L12, L18, M11, M13, M15 M17, M19, N8, N10, N12, N14, N16, N18, N20, N22, P9, P11, P13, P15, P17, P19, P21, R8, R10, R12, R14, R16, R18, R20, R22, T9, T11, T13, T15, T17, T19, T21, U8, U10, U12, U14, U16, U18, U20, U22, V9, V11, V13, V15, V17, V19, V21, V23, W8, W10, W18, W20, W22, Y9, Y19, Y21, Y23, AA8, AA10, AA12, AA14, AA16, AA18, AA20, AA22, AB23	0.9 to 1.1	SmartReflex core supply voltage
CVDD1	G6, H1, H3, H5, H7, H9, J2, J4, J6, J8, J10, J14, J16, J20, J22, K7, K9, K13, K15, K17, K21, L8, L10, L14, L16, L20, L22, M9, M21, W12, W14, W16, Y11, Y13, Y15, Y17, AD1, AD3, AE2, AF1, AG2, AH1, AJ2	1.0	Fixed core supply voltage
DVDD15	A2, A11, A17, A28, B1, B29, C14, C25, D5, D8, D20, D23, E3, F5, F7, F9, F11, F17, F19, F27, G2, G4, G8, G10, G12, G14, G16 G18, G20, G22, G24	1.5	DDR IO supply
DVDD18	G28, H25, V5, Y5, Y25, AB5, AB17, AB19, AB21, AC2, AC4, AE24, AE27, AF19, AF22, AH26, AH29, AJ28	1.8	IO supply
VDDR1	K6	1.5	HyperLink SerDes regulator supply
VDDR2	AE15	1.5	PCIe SerDes regulator supply
VDDR3	AE6	1.5	SGMII SerDes regulator supply
VDDR4	AE11	1.5	SRIO SerDes regulator supply
VDDR5	R25	1.5	AIF SerDes regulator supply
VDDR6	N25		
VDDT1	M7, N6, P7, R6, T7, V7, W6, Y7	1.0	HyperLink SerDes termination supply
VDDT2	AC6, AC8, AC10, AC12, AC14, AD5, AD7, AD9, AD11, AD13, AE4, AE10, AE12	1.0	SGMII/SRIO/PCIe SerDes termination supply
VDDT3	K25, L24, M23, M25, N24, P23, P25, R24, T23, T25, U24, V25	1.0	AIF SerDes termination supply
VREFSSTL	E14	0.75	DDR3 reference voltage
VSS	A1, A29, B11, B17, B25, C8, C23, D3, D14, D18, E5, E20, F6, F8, F10, F12, F16, F18, F26, F28, F29, G1, G3, G5, G7, G9, G11, G13, G15, G17, G19, G21, G23, G25, H2, H4, H6, H8, H10, H12, H14, H16, H18, H20, H22, J1, J3, J5, J7, J9, J11, J13, J15, J17, J19, J21, J27, J29, K1, K2, K3, K4, K5, K8, K10, K12, K14, K16, K18, K20, K22, K26, K28, L2, L3,L5, L6, L7, L9, L11, L13, L15, L17, L19, L21, L23, M3, M6, M8, M10, M12, M14, M16, M18, M20, M22, M24, M27, M29, N1, N3, N4, N7, N9, N11, N13, N15, N17, N19, N21, N23, N26, N28, P2, P3, P5, P6, P8, P10, P12, P14, P16, P18, P20, P22, P24, R3, R7, R9, R11, R13, R15, R17, R19, R21, R23, R27, R29, T1, T3, T4, T6, T8, T10, T12, T14, T16, T18, T20, T22, T24, T26, T28, U1, U2, U3, U4, U5, U6, U7, U9, U11, U13, U15, U17, U19, U21, U23, V6, V8, V10, V12, V14, V16, V18, V20, V22, V24, V27, V29, W5, W7, W9, W11, W13, W15, W17, W19, W21, W23, W25, W26, W28, W29, Y6, Y8, Y10, Y12, Y14, Y16, Y18, Y20, Y22, Y24, Y26, AA5, AA6, AA7, AA9, AA11, AA13, AA15, AA17, AA19, AA21, AA23, AA25, AB2, AB6, AB7, AB8, AB9, AB10, AB11, AB12, AB13, AB14, AB16, AB18, AB20, AB22, AB24, AC1, AC3, AC5, AC7, AC9, AC11, AC13, AC15, AD2, AD4, AD6, AD8, AD10, AD12, AD14, AD24, AD27, AE1, AE3, AE7, AE9, AE13, AF2, AF5, AF8, AF11, AF14, AF15, AF20, AG1, AG3, AG6, AG9, AG12, AG15, AG22, AG26, AH2, AH5, AH8, AH11, AH14, AJ1, AJ3, AJ6, AJ9, AJ12, AJ15, AJ29	Gnd	Ground
End of Table 2-16			

Table 2-17 Terminal Functions — By Signal Name (Part 1 of 11)

Signal Name	Ball Number
AIFRXN0	L28
AIFRXN1	K29
AIFRXN2	R28
AIFRXN3	P29
AIFRXN4	T29
AIFRXN5	U28
AIFRXP0	M28
AIFRXP1	L29
AIFRXP2	P28
AIFRXP3	N29
AIFRXP4	U29
AIFRXP5	V28
AIFTXN0	L26
AIFTXN1	L27
AIFTXN2	R26
AIFTXN3	P27
AIFTXN4	U27
AIFTXN5	U26
AIFTXP0	M26
AIFTXP1	K27
AIFTXP2	P26
AIFTXP3	N27
AIFTXP4	T27
AIFTXP5	V26
ALTCORECLKN	AB28
ALTCORECLKP	AB29
AVDDA1	W24
AVDDA2	J26
AVDDA3	AB15
BOOTCOMPLETE	AC21
BOOTMODE00 †	AG18
BOOTMODE01 †	AD19
BOOTMODE02 †	AE19
BOOTMODE03 †	AF18
BOOTMODE04 †	AE18
BOOTMODE05 †	AG20
BOOTMODE06 †	AH19
BOOTMODE07 †	AJ19
BOOTMODE08 †	AE21
BOOTMODE09 †	AG19
BOOTMODE10 †	AD20
BOOTMODE11 †	AE20

Table 2-17 Terminal Functions — By Signal Name (Part 2 of 11)

Signal Name	Ball Number
BOOTMODE12 †	AF21
CORECLKSEL	AB25
CORESEL0	AH15
CORESEL1	AC16
CORESEL2	AD15
CVDD	H11, H13, H15, H17, H19, H21, J12, J18, K11, K19, L12, L18, M11, M13, M15 M17, M19, N8, N10, N12, N14, N16, N18, N20, N22, P9, P11, P13, P15, P17, P19
CVDD	P21, R8, R10, R12, R14, R16, R18, R20, R22, T9, T11, T13, T15, T17, T19, T21, U8, U10, U12, U14, U16, U18, U20, U22, V9, V11, V13, V15, V17, V19, V21, V23
CVDD	W8, W10, W18, W20, W22, Y9, Y19, Y21, Y23, AA8, AA10, AA12, AA14, AA16, AA18, AA20, AA22, AB23
CVDD1	G6, H1, H3, H5, H7, H9, J2, J4, J6, J8, J10, J14, J16, J20, J22, K7, K9, K13, K15, K17, K21, L8, L10, L14, L16, L20, L22, M9, M21, W12, W14, W16, Y11, Y13, Y15, Y17, AD1, AD3, AE2, AF1, AG2, AH1, AJ2
DDRA00	A14
DDRA01	B14
DDRA02	F14
DDRA03	F13
DDRA04	A15
DDRA05	C15
DDRA06	B15
DDRA07	D15
DDRA08	F15
DDRA09	E15
DDRA10	E16
DDRA11	D16
DDRA12	E17
DDRA13	C16
DDRA14	D17

Table 2-17 Terminal Functions — By Signal Name (Part 3 of 11)

Signal Name	Ball Number
DDRA15	C17
DDRBA0	A13
DDRBA1	B13
DDRBA2	C13
$\overline{\text{DDRCAS}}$	D12
DDRCB00	E19
DDRCB01	C20
DDRCB02	D19
DDRCB03	B20
DDRCB04	C19
DDRCB05	C18
DDRCB06	B18
DDRCB07	A18
$\overline{\text{DDRCE0}}$	C11
$\overline{\text{DDRCE1}}$	C12
DDRCKE0	D11
DDRCKE1	E18
DDRCLKN	H29
DDRCLKOUTN0	B12
DDRCLKOUTN1	B16
DDRCLKOUTP0	A12
DDRCLKOUTP1	A16
DDRCLKP	G29
DDRD00	E28
DDRD01	D29
DDRD02	E27
DDRD03	D28
DDRD04	D27
DDRD05	B28
DDRD06	E26
DDRD07	F25
DDRD08	F24
DDRD09	E24
DDRD10	E25
DDRD11	D25
DDRD12	D26
DDRD13	C26
DDRD14	B26
DDRD15	A26
DDRD16	F23
DDRD17	F22
DDRD18	D24

Table 2-17 Terminal Functions — By Signal Name (Part 4 of 11)

Signal Name	Ball Number
DDRD19	E23
DDRD20	A23
DDRD21	B23
DDRD22	C24
DDRD23	E22
DDRD24	D21
DDRD25	F20
DDRD26	E21
DDRD27	F21
DDRD28	D22
DDRD29	C21
DDRD30	B22
DDRD31	C22
DDRD32	E10
DDRD33	D10
DDRD34	B10
DDRD35	D9
DDRD36	E9
DDRD37	C9
DDRD38	B8
DDRD39	E8
DDRD40	A7
DDRD41	D7
DDRD42	E7
DDRD43	C7
DDRD44	B7
DDRD45	E6
DDRD46	D6
DDRD47	C6
DDRD48	C5
DDRD49	A5
DDRD50	B4
DDRD51	A4
DDRD52	D4
DDRD53	E4
DDRD54	C4
DDRD55	C3
DDRD56	F4
DDRD57	D2
DDRD58	E2
DDRD59	C2
DDRD60	F2

Table 2-17 Terminal Functions — By Signal Name (Part 5 of 11)

Signal Name	Ball Number
DDRD61	F3
DDRD62	E1
DDRD63	F1
DDRDQM0	E29
DDRDQM1	C27
DDRDQM2	A25
DDRDQM3	A22
DDRDQM4	A10
DDRDQM5	A8
DDRDQM6	B5
DDRDQM7	B2
DDRDQM8	A20
DDRDQS0N	C29
DDRDQS0P	C28
DDRDQS1N	B27
DDRDQS1P	A27
DDRDQS2N	B24
DDRDQS2P	A24
DDRDQS3N	B21
DDRDQS3P	A21
DDRDQS4N	B9
DDRDQS4P	A9
DDRDQS5N	A6
DDRDQS5P	B6
DDRDQS6N	A3
DDRDQS6P	B3
DDRDQS7N	C1
DDRDQS7P	D1
DDRDQS8N	B19
DDRDQS8P	A19
DDRODT0	D13
DDRODT1	E13
$\overline{\text{DDRRAS}}$	C10
$\overline{\text{DDRRESET}}$	E11
DDRSLRATE0	H27
DDRSLRATE1	H26
$\overline{\text{DDRWE}}$	E12
DVDD15	A2, A11, A17, A28, B1, B29, C14, C25, D5, D8, D20, D23, E3, F5, F7, F9, F11, F17, F19, F27, G2, G4, G8, G10, G12, G14, G16 G18, G20, G22, G24

Table 2-17 Terminal Functions — By Signal Name (Part 6 of 11)

Signal Name	Ball Number
DVDD18	G28, H25, V5, Y5, Y25, AB5, AB17, AB19, AB21, AC2, AC4, AE24, AE27, AF19, AF22, AH26, AH29, AJ28
EMU00	AE29
EMU01	AF29
EMU02	AE28
EMU03	AF28
EMU04	AE26
EMU05	AD25
EMU06	AF25
EMU07	AE25
EMU08	AF27
EMU09	AG29
EMU10	AF26
EMU11	AG28
EMU12	AG27
EMU13	AG25
EMU14	AH28
EMU15	AJ27
EMU16	AH27
EMU17	AJ26
EMU18	AH25
EXTFRAMEEVENT	AE17
GPIO00	AJ20
GPIO01	AG18
GPIO02	AD19
GPIO03	AE19
GPIO04	AF18
GPIO05	AE18
GPIO06	AG20
GPIO07	AH19
GPIO08	AJ19
GPIO09	AE21
GPIO10	AG19
GPIO11	AD20
GPIO12	AE20
GPIO13	AF21
GPIO14	AH20
GPIO15	AD21
HOUT	AC18
LENDIAN	AJ20 †

Table 2-17 Terminal Functions — By Signal Name (Part 7 of 11)

Signal Name	Ball Number
LRESETNMIEN	AC20
LRESET	AE22
MCMCLKN	W2
MCMCLKP	W1
MCMREFCLKOUTN	V1
MCMREFCLKOUTP	V2
MCMRXFLCLK	V3
MCMRXFLDAT	W3
MCMRXN0	T2
MCMRXN1	P1
MCMRXN2	L1
MCMRXN3	N2
MCMRXP0	R2
MCMRXP1	R1
MCMRXP2	M1
MCMRXP3	M2
MCMRXPMCLK	AA3
MCMRXPMDAT	Y3
MCMTXFLCLK	Y1
MCMTXFLDAT	Y2
MCMTXN0	T5
MCMTXN1	R4
MCMTXN2	L4
MCMTXN3	M5
MCMTXP0	R5
MCMTXP1	P4
MCMTXP2	M4
MCMTXP3	N5
MCMTXPMCLK	AA2
MCMTXPMDAT	AA1
MDCLK	AF16
MDIO	AG16
NMI	AC25
PACLKSEL	AD23
PASSCLKN	AH18
PASSCLKP	AJ18
PCIECLKN	AJ17
PCIECLKP	AH17
PCIERXN0	AJ14
PCIERXN1	AH12
PCIERXP0	AJ13
PCIERXP1	AH13

Table 2-17 Terminal Functions — By Signal Name (Part 8 of 11)

Signal Name	Ball Number
PCIESSMODE0 †	AH20
PCIESSMODE1 †	AD21
PCIESSEN †	AJ23
PCIETXN0	AG14
PCIETXN1	AF12
PCIETXP0	AG13
PCIETXP1	AF13
PHYSYNC	AB27
POR	AC19
PTV15	H24
RADSYNC	AA27
RESETFULL	AE23
RESETSTAT	AD18
RESET	AC24
RIORXN0	AJ11
RIORXN1	AH10
RIORXN2	AJ7
RIORXN3	AH6
RIORXP0	AJ10
RIORXP1	AH9
RIORXP2	AJ8
RIORXP3	AH7
RIOTXN0	AG11
RIOTXN1	AF9
RIOTXN2	AG7
RIOTXN3	AF6
RIOTXP0	AG10
RIOTXP1	AF10
RIOTXP2	AG8
RIOTXP3	AF7
RP1CLKN	AA28
RP1CLKP	Y28
RP1FBN	AA29
RP1FBP	Y29
RSV01	AJ25
RSV03	AC23
RSV04	Y27
RSV05	W27
RSV06	J28
RSV07	H28
RSV08	J24
RSV09	J25

Table 2-17 Terminal Functions — By Signal Name (Part 9 of 11)

Signal Name	Ball Number
RSV0A	K24
RSV0B	K23
RSV10	H23
RSV11	J23
RSV12	AD22
RSV13	AC22
RSV14	V4
RSV15	AE8
RSV16	AE14
RSV17	AE5
RSV18	AA24
RSV19	G27
RSV20	AB26
RSV21	G26
RSV22	AE16
RSV23	AD16
RSV24	AG17
RSV25	AF17
RSV26	U25
RSV27	L25
RSV28	Y4
RSV29	W4
SCL	AC17
SDA	AD17
SGMII0RXN	AH3
SGMII0RXP	AH4
SGMII0TXN	AF3
SGMII0TXP	AF4
SGMII1RXN	AJ4
SGMII1RXP	AJ5
SGMII1TXN	AG4
SGMII1TXP	AG5
SPICLK	AG21
SPIDIN	AH22
SPIDOUT	AJ21
SPISCS0	AH21
SPISCS1	AJ22
SRIOSGMIICLKN	AH16
SRIOSGMIICLKP	AJ16
SYSCLKN	AC28
SYSCLKOUT	AA26
SYSCLKP	AC29

Table 2-17 Terminal Functions — By Signal Name (Part 10 of 11)

Signal Name	Ball Number
TCK	AD29
TDI	AD28
TDO	AC27
TIMI0	AJ23
TIMI1	AG23
TIMO0	AH23
TIMO1	AF23
TMS	AC26
$\overline{\text{TRST}}$	AD26
UARTCTS	AH24
UARTRTS	AG24
UARTRXD	AF24
UARTTXD	AJ24
VCNTL0	AB4
VCNTL1	AB3
VCNTL2	AA4
VCNTL3	AB1
VDDR1	K6
VDDR2	AE15
VDDR3	AE6
VDDR4	AE11
VDDR5	R25
VDDR6	N25
VDDT1	M7, N6, P7, R6, T7, V7, W6, Y7
VDDT2	AC6, AC8, AC10, AC12, AC14, AD5, AD7, AD9, AD11, AD13, AE4, AE10, AE12
VDDT3	K25, L24, M23, M25, N24, P23, P25, R24, T23, T25, U24, V25
VREFSSTL	E14
VSS	A1, A29, B11, B17, B25, C8, C23, D3, D14, D18, E5, E20, F6, F8, F10, F12, F16, F18, F26, F28, F29, G1, G3, G5, G7, G9, G11, G13, G15, G17, G19, G21, G23, G25
VSS	H2, H4, H6, H8, H10, H12, H14, H16, H18, H20, H22, J1, J3, J5, J7, J9, J11, J13, J15, J17, J19, J21, J27, J29, K1, K2, K3, K4, K5, K8, K10, K12, K14, K16, K18, K20

Table 2-17 Terminal Functions — By Signal Name (Part 11 of 11)

Signal Name	Ball Number
VSS	K22, K26, K28, L2, L3,L5, L6, L7, L9, L11, L13, L15, L17, L19, L21, L23, M3, M6, M8, M10, M12, M14, M16, M18, M20, M22, M24, M27, M29, N1, N3, N4, N7
VSS	N9, N11, N13, N15, N17, N19, N21, N23, N26, N28, P2, P3, P5, P6, P8, P10, P12, P14, P16, P18, P20, P22, P24, R3, R7, R9, R11, R13, R15, R17, R19, R21, R23, R27
VSS	R29, T1, T3, T4, T6, T8, T10, T12, T14, T16, T18, T20, T22, T24, T26, T28, U1, U2, U3, U4, U5, U6, U7, U9, U11, U13, U15, U17, U19, U21, U23, V6, V8, V10
VSS	V12, V14, V16, V18, V20, V22, V24, V27, V29, W5, W7, W9, W11, W13, W15, W17, W19, W21, W23, W25, W26, W28, W29, Y6, Y8, Y10, Y12, Y14, Y16
VSS	Y18, Y20, Y22, Y24, Y26, AA5, AA6, AA7, AA9, AA11, AA13, AA15, AA17, AA19, AA21, AA23, AA25, AB2, AB6, AB7, AB8, AB9, AB10, AB11, AB12, AB13, AB14
VSS	AB16, AB18, AB20, AB22, AB24, AC1, AC3, AC5, AC7, AC9, AC11, AC13, AC15, AD2, AD4, AD6, AD8, AD10, AD12, AD14, AD24, AD27, AE1, AE3, AE7, AE9
VSS	AE13, AF2, AF5, AF8, AF11, AF14, AF15, AF20, AG1, AG3, AG6, AG9, AG12, AG15, AG22, AG26, AH2, AH5, AH8, AH11, AH14, AJ1, AJ3, AJ6, AJ9, AJ12
VSS	AJ15, AJ29
End of Table 2-17	

Table 2-18 Terminal Functions — By Ball Number (Part 1 of 21)

Ball Number	Signal Name
A1	VSS
A2	DVDD15
A3	DDRDQS6N
A4	DDRD51
A5	DDRD49
A6	DDRDQS5N
A7	DDRD40
A8	DDRDQM5
A9	DDRDQS4P
A10	DDRDQM4
A11	DVDD15
A12	DDRCLKOUTP0
A13	DDRBA0
A14	DDRA00
A15	DDRA04
A16	DDRCLKOUTP1
A17	DVDD15
A18	DDRCB07
A19	DDRDQS8P
A20	DDRDQM8
A21	DDRDQS3P
A22	DDRDQM3
A23	DDRD20
A24	DDRDQS2P
A25	DDRDQM2
A26	DDRD15
A27	DDRDQS1P
A28	DVDD15
A29	VSS
B1	DVDD15
B2	DDRDQM7
B3	DDRDQS6P
B4	DDRD50
B5	DDRDQM6
B6	DDRDQS5P
B7	DDRD44
B8	DDRD38
B9	DDRDQS4N
B10	DDRD34
B11	VSS
B12	DDRCLKOUTN0
B13	DDRBA1

Table 2-18 Terminal Functions — By Ball Number (Part 2 of 21)

Ball Number	Signal Name
B14	DDRA01
B15	DDRA06
B16	DDRCLKOUTN1
B17	VSS
B18	DDRCB06
B19	DDRDQS8N
B20	DDRCB03
B21	DDRDQS3N
B22	DDRD30
B23	DDRD21
B24	DDRDQS2N
B25	VSS
B26	DDRD14
B27	DDRDQS1N
B28	DDRD05
B29	DVDD15
C1	DDRDQS7N
C2	DDRD59
C3	DDRD55
C4	DDRD54
C5	DDRD48
C6	DDRD47
C7	DDRD43
C8	VSS
C9	DDRD37
C10	$\overline{\text{DDRRAS}}$
C11	$\overline{\text{DDRCE0}}$
C12	$\overline{\text{DDRCE1}}$
C13	DDRBA2
C14	DVDD15
C15	DDRA05
C16	DDRA13
C17	DDRA15
C18	DDRCB05
C19	DDRCB04
C20	DDRCB01
C21	DDRD29
C22	DDRD31
C23	VSS
C24	DDRD22
C25	DVDD15
C26	DDRD13

Table 2-18 Terminal Functions — By Ball Number (Part 3 of 21)

Ball Number	Signal Name
C27	DDRDQM1
C28	DDRDQS0P
C29	DDRDQS0N
D1	DDRDQS7P
D2	DDRD57
D3	VSS
D4	DDRD52
D5	DVDD15
D6	DDRD46
D7	DDRD41
D8	DVDD15
D9	DDRD35
D10	DDRD33
D11	DDRCKE0
D12	$\overline{\text{DDRCAS}}$
D13	DDRODT0
D14	VSS
D15	DDRA07
D16	DDRA11
D17	DDRA14
D18	VSS
D19	DDRCB02
D20	DVDD15
D21	DDRD24
D22	DDRD28
D23	DVDD15
D24	DDRD18
D25	DDRD11
D26	DDRD12
D27	DDRD04
D28	DDRD03
D29	DDRD01
E1	DDRD62
E2	DDRD58
E3	DVDD15
E4	DDRD53
E5	VSS
E6	DDRD45
E7	DDRD42
E8	DDRD39
E9	DDRD36
E10	DDRD32

Table 2-18 Terminal Functions — By Ball Number (Part 4 of 21)

Ball Number	Signal Name
E11	$\overline{\text{DDRRESET}}$
E12	$\overline{\text{DDRWE}}$
E13	DDRODT1
E14	VREFSSTL
E15	DDRA09
E16	DDRA10
E17	DDRA12
E18	DDRCKE1
E19	DDRCB00
E20	VSS
E21	DDRD26
E22	DDRD23
E23	DDRD19
E24	DDRD09
E25	DDRD10
E26	DDRD06
E27	DDRD02
E28	DDRD00
E29	DDRDQM0
F1	DDRD63
F2	DDRD60
F3	DDRD61
F4	DDRD56
F5	DVDD15
F6	VSS
F7	DVDD15
F8	VSS
F9	DVDD15
F10	VSS
F11	DVDD15
F12	VSS
F13	DDRA03
F14	DDRA02
F15	DDRA08
F16	VSS
F17	DVDD15
F18	VSS
F19	DVDD15
F20	DDRD25
F21	DDRD27
F22	DDRD17
F23	DDRD16

Table 2-18 Terminal Functions — By Ball Number (Part 5 of 21)

Ball Number	Signal Name
F24	DDRD08
F25	DDRD07
F26	VSS
F27	DVDD15
F28	VSS
F29	VSS
G1	VSS
G2	DVDD15
G3	VSS
G4	DVDD15
G5	VSS
G6	CVDD1
G7	VSS
G8	DVDD15
G9	VSS
G10	DVDD15
G11	VSS
G12	DVDD15
G13	VSS
G14	DVDD15
G15	VSS
G16	DVDD15
G17	VSS
G18	DVDD15
G19	VSS
G20	DVDD15
G21	VSS
G22	DVDD15
G23	VSS
G24	DVDD15
G25	VSS
G26	RSV21
G27	RSV19
G28	DVDD18
G29	DDRCLKP
H1	CVDD1
H2	VSS
H3	CVDD1
H4	VSS
H5	CVDD1
H6	VSS
H7	CVDD1

Table 2-18 Terminal Functions — By Ball Number (Part 6 of 21)

Ball Number	Signal Name
H8	VSS
H9	CVDD1
H10	VSS
H11	CVDD
H12	VSS
H13	CVDD
H14	VSS
H15	CVDD
H16	VSS
H17	CVDD
H18	VSS
H19	CVDD
H20	VSS
H21	CVDD
H22	VSS
H23	RSV10
H24	PTV15
H25	DVDD18
H26	DDRSLRATE1
H27	DDRSLRATE0
H28	RSV07
H29	DDRCLKN
J1	VSS
J2	CVDD1
J3	VSS
J4	CVDD1
J5	VSS
J6	CVDD1
J7	VSS
J8	CVDD1
J9	VSS
J10	CVDD1
J11	VSS
J12	CVDD
J13	VSS
J14	CVDD1
J15	VSS
J16	CVDD1
J17	VSS
J18	CVDD
J19	VSS
J20	CVDD1

Table 2-18 Terminal Functions — By Ball Number (Part 7 of 21)

Ball Number	Signal Name
J21	VSS
J22	CVDD1
J23	RSV11
J24	RSV08
J25	RSV09
J26	AVDDA2
J27	VSS
J28	RSV06
J29	VSS
K1	VSS
K2	VSS
K3	VSS
K4	VSS
K5	VSS
K6	VDDR1
K7	CVDD1
K8	VSS
K9	CVDD1
K10	VSS
K11	CVDD
K12	VSS
K13	CVDD1
K14	VSS
K15	CVDD1
K16	VSS
K17	CVDD1
K18	VSS
K19	CVDD
K20	VSS
K21	CVDD1
K22	VSS
K23	RSV0B
K24	RSV0A
K25	VDDT3
K26	VSS
K27	AIFTXP1
K28	VSS
K29	AIFRXN1
L1	MCMRXN2
L2	VSS
L3	VSS
L4	MCMTXN2

Table 2-18 Terminal Functions — By Ball Number (Part 8 of 21)

Ball Number	Signal Name
L5	VSS
L6	VSS
L7	VSS
L8	CVDD1
L9	VSS
L10	CVDD1
L11	VSS
L12	CVDD
L13	VSS
L14	CVDD1
L15	VSS
L16	CVDD1
L17	VSS
L18	CVDD
L19	VSS
L20	CVDD1
L21	VSS
L22	CVDD1
L23	VSS
L24	VDDT3
L25	RSV27
L26	AIFTXN0
L27	AIFTXN1
L28	AIFRXN0
L29	AIFRXP1
M1	MCMRXP2
M2	MCMRXP3
M3	VSS
M4	MCMTXP2
M5	MCMTXN3
M6	VSS
M7	VDDT1
M8	VSS
M9	CVDD1
M10	VSS
M11	CVDD
M12	VSS
M13	CVDD
M14	VSS
M15	CVDD
M16	VSS
M17	CVDD

Table 2-18 Terminal Functions — By Ball Number (Part 9 of 21)

Ball Number	Signal Name
M18	VSS
M19	CVDD
M20	VSS
M21	CVDD1
M22	VSS
M23	VDDT3
M24	VSS
M25	VDDT3
M26	AIFTXP0
M27	VSS
M28	AIFRXP0
M29	VSS
N1	VSS
N2	MCMRXN3
N3	VSS
N4	VSS
N5	MCMTXP3
N6	VDDT1
N7	VSS
N8	CVDD
N9	VSS
N10	CVDD
N11	VSS
N12	CVDD
N13	VSS
N14	CVDD
N15	VSS
N16	CVDD
N17	VSS
N18	CVDD
N19	VSS
N20	CVDD
N21	VSS
N22	CVDD
N23	VSS
N24	VDDT3
N25	VDDR6
N26	VSS
N27	AIFTXP3
N28	VSS
N29	AIFRXP3
P1	MCMRXN1

Table 2-18 Terminal Functions — By Ball Number (Part 10 of 21)

Ball Number	Signal Name
P2	VSS
P3	VSS
P4	MCMTXP1
P5	VSS
P6	VSS
P7	VDDT1
P8	VSS
P9	CVDD
P10	VSS
P11	CVDD
P12	VSS
P13	CVDD
P14	VSS
P15	CVDD
P16	VSS
P17	CVDD
P18	VSS
P19	CVDD
P20	VSS
P21	CVDD
P22	VSS
P23	VDDT3
P24	VSS
P25	VDDT3
P26	AIFTXP2
P27	AIFTXN3
P28	AIFRXP2
P29	AIFRXN3
R1	MCMRXP1
R2	MCMRXP0
R3	VSS
R4	MCMTXN1
R5	MCMTXP0
R6	VDDT1
R7	VSS
R8	CVDD
R9	VSS
R10	CVDD
R11	VSS
R12	CVDD
R13	VSS
R14	CVDD

Table 2-18 Terminal Functions — By Ball Number (Part 11 of 21)

Ball Number	Signal Name
R15	VSS
R16	CVDD
R17	VSS
R18	CVDD
R19	VSS
R20	CVDD
R21	VSS
R22	CVDD
R23	VSS
R24	VDDT3
R25	VDDR5
R26	AIFTXN2
R27	VSS
R28	AIFRXN2
R29	VSS
T1	VSS
T2	MCMRXN0
T3	VSS
T4	VSS
T5	MCMTXN0
T6	VSS
T7	VDDT1
T8	VSS
T9	CVDD
T10	VSS
T11	CVDD
T12	VSS
T13	CVDD
T14	VSS
T15	CVDD
T16	VSS
T17	CVDD
T18	VSS
T19	CVDD
T20	VSS
T21	CVDD
T22	VSS
T23	VDDT3
T24	VSS
T25	VDDT3
T26	VSS
T27	AIFTXP4

Table 2-18 Terminal Functions — By Ball Number (Part 12 of 21)

Ball Number	Signal Name
T28	VSS
T29	AIFRXN4
U1	VSS
U2	VSS
U3	VSS
U4	VSS
U5	VSS
U6	VSS
U7	VSS
U8	CVDD
U9	VSS
U10	CVDD
U11	VSS
U12	CVDD
U13	VSS
U14	CVDD
U15	VSS
U16	CVDD
U17	VSS
U18	CVDD
U19	VSS
U20	CVDD
U21	VSS
U22	CVDD
U23	VSS
U24	VDDT3
U25	RSV26
U26	AIFTXN5
U27	AIFTXN4
U28	AIFRXN5
U29	AIFRXP4
V1	MCMREFCLKOUTN
V2	MCMREFCLKOUTP
V3	MCMRXFLCLK
V4	RSV14
V5	DVDD18
V6	VSS
V7	VDDT1
V8	VSS
V9	CVDD
V10	VSS
V11	CVDD

Table 2-18 Terminal Functions — By Ball Number (Part 13 of 21)

Ball Number	Signal Name
V12	VSS
V13	CVDD
V14	VSS
V15	CVDD
V16	VSS
V17	CVDD
V18	VSS
V19	CVDD
V20	VSS
V21	CVDD
V22	VSS
V23	CVDD
V24	VSS
V25	VDDT3
V26	AIFTXP5
V27	VSS
V28	AIFRXP5
V29	VSS
W1	MCMCLKP
W2	MCMCLKN
W3	MCMRXFLDAT
W4	RSV29
W5	VSS
W6	VDDT1
W7	VSS
W8	CVDD
W9	VSS
W10	CVDD
W11	VSS
W12	CVDD1
W13	VSS
W14	CVDD1
W15	VSS
W16	CVDD1
W17	VSS
W18	CVDD
W19	VSS
W20	CVDD
W21	VSS
W22	CVDD
W23	VSS
W24	AVDDA1

Table 2-18 Terminal Functions — By Ball Number (Part 14 of 21)

Ball Number	Signal Name
W25	VSS
W26	VSS
W27	RSV05
W28	VSS
W29	VSS
Y1	MCMTXFLCLK
Y2	MCMTXFLDAT
Y3	MCMRXPMDAT
Y4	RSV28
Y5	DVDD18
Y6	VSS
Y7	VDDT1
Y8	VSS
Y9	CVDD
Y10	VSS
Y11	CVDD1
Y12	VSS
Y13	CVDD1
Y14	VSS
Y15	CVDD1
Y16	VSS
Y17	CVDD1
Y18	VSS
Y19	CVDD
Y20	VSS
Y21	CVDD
Y22	VSS
Y23	CVDD
Y24	VSS
Y25	DVDD18
Y26	VSS
Y27	RSV04
Y28	RP1CLKP
Y29	RP1FBP
AA1	MCMTXPMDAT
AA2	MCMTXPMCLK
AA3	MCMRXPMCLK
AA4	VCNTL2
AA5	VSS
AA6	VSS
AA7	VSS
AA8	CVDD

Table 2-18 Terminal Functions — By Ball Number (Part 15 of 21)

Ball Number	Signal Name
AA9	VSS
AA10	CVDD
AA11	VSS
AA12	CVDD
AA13	VSS
AA14	CVDD
AA15	VSS
AA16	CVDD
AA17	VSS
AA18	CVDD
AA19	VSS
AA20	CVDD
AA21	VSS
AA22	CVDD
AA23	VSS
AA24	RSV18
AA25	VSS
AA26	SYSCLKOUT
AA27	RADSYNC
AA28	RP1CLKN
AA29	RP1FBN
AB1	VCNTL3
AB2	VSS
AB3	VCNTL1
AB4	VCNTL0
AB5	DVDD18
AB6	VSS
AB7	VSS
AB8	VSS
AB9	VSS
AB10	VSS
AB11	VSS
AB12	VSS
AB13	VSS
AB14	VSS
AB15	AVDDA3
AB16	VSS
AB17	DVDD18
AB18	VSS
AB19	DVDD18
AB20	VSS
AB21	DVDD18

Table 2-18 Terminal Functions — By Ball Number (Part 16 of 21)

Ball Number	Signal Name
AB22	VSS
AB23	CVDD
AB24	VSS
AB25	CORECLKSEL
AB26	RSV20
AB27	PHYSYNC
AB28	ALTCORECLKN
AB29	ALTCORECLKP
AC1	VSS
AC2	DVDD18
AC3	VSS
AC4	DVDD18
AC5	VSS
AC6	VDDT2
AC7	VSS
AC8	VDDT2
AC9	VSS
AC10	VDDT2
AC11	VSS
AC12	VDDT2
AC13	VSS
AC14	VDDT2
AC15	VSS
AC16	CORESEL1
AC17	SCL
AC18	HOUT
AC19	$\overline{\text{POR}}$
AC20	$\overline{\text{LRESETNMIEN}}$
AC21	BOOTCOMPLETE
AC22	RSV13
AC23	RSV03
AC24	$\overline{\text{RESET}}$
AC25	$\overline{\text{NMI}}$
AC26	TMS
AC27	TDO
AC28	SYSCLKN
AC29	SYSCLKP
AD1	CVDD1
AD2	VSS
AD3	CVDD1
AD4	VSS
AD5	VDDT2

Table 2-18 Terminal Functions — By Ball Number (Part 17 of 21)

Ball Number	Signal Name
AD6	VSS
AD7	VDDT2
AD8	VSS
AD9	VDDT2
AD10	VSS
AD11	VDDT2
AD12	VSS
AD13	VDDT2
AD14	VSS
AD15	CORESEL2
AD16	RSV23
AD17	SDA
AD18	$\overline{\text{RESETSTAT}}$
AD19	GPIO02
AD19 †	BOOTMODE01
AD20	GPIO11
AD20 †	BOOTMODE10
AD21	GPIO15
AD21 †	PCIESSMODE1
AD22	RSV12
AD23	PACLKSEL
AD24	VSS
AD25	EMU05
AD26	$\overline{\text{TRST}}$
AD27	VSS
AD28	TDI
AD29	TCK
AE1	VSS
AE2	CVDD1
AE3	VSS
AE4	VDDT2
AE5	RSV17
AE6	VDDR3
AE7	VSS
AE8	RSV15
AE9	VSS
AE10	VDDT2
AE11	VDDR4
AE12	VDDT2
AE13	VSS
AE14	RSV16
AE15	VDDR2

Table 2-18 Terminal Functions — By Ball Number (Part 18 of 21)

Ball Number	Signal Name
AE16	RSV22
AE17	EXTFRAMEEVENT
AE18	GPIO05
AE18 †	BOOTMODE04
AE19	GPIO03
AE19 †	BOOTMODE02
AE20	GPIO12
AE20 †	BOOTMODE11
AE21	GPIO09
AE22	$\overline{\text{LRESET}}$
AE23	$\overline{\text{RESETFULL}}$
AE24	DVDD18
AE25	EMU07
AE26	EMU04
AE27	DVDD18
AE28	EMU02
AE29	EMU00
AF1	CVDD1
AF2	VSS
AF3	SGMII0TXN
AF4	SGMII0TXP
AF5	VSS
AF6	RIOTXN3
AF7	RIOTXP3
AF8	VSS
AF9	RIOTXN1
AF10	RIOTXP1
AF11	VSS
AF12	PCIETXN1
AF13	PCIETXP1
AF14	VSS
AF15	VSS
AF16	MDCLK
AF17	RSV25
AF18	GPIO04
AF18 †	BOOTMODE03
AF19	DVDD18
AF20	VSS
AF21	GPIO13
AF21 †	BOOTMODE12
AF22	DVDD18
AF23	TIMO1

Table 2-18 Terminal Functions — By Ball Number (Part 19 of 21)

Ball Number	Signal Name
AF24	UARTRXD
AF25	EMU06
AF26	EMU10
AF27	EMU08
AF28	EMU03
AF29	EMU01
AG1	VSS
AG2	CVDD1
AG3	VSS
AG4	SGMII1TXN
AG5	SGMII1TXP
AG6	VSS
AG7	RIOTXN2
AG8	RIOTXP2
AG9	VSS
AG10	RIOTXP0
AG11	RIOTXN0
AG12	VSS
AG13	PCIETXP0
AG14	PCIETXN0
AG15	VSS
AG16	MDIO
AG17	RSV24
AG18	GPIO01
AG18 †	BOOTMODE00
AG19	GPIO10
AG20	GPIO06
AG20 †	BOOTMODE05
AG21	SPICLK
AG22	VSS
AG23	TIMI1
AG24	UARTRTS
AG25	EMU13
AG26	VSS
AG27	EMU12
AG28	EMU11
AG29	EMU09
AH1	CVDD1
AH2	VSS
AH3	SGMII0RXN
AH4	SGMII0RXP
AH5	VSS

Table 2-18 Terminal Functions — By Ball Number (Part 20 of 21)

Ball Number	Signal Name
AH6	RIORXN3
AH7	RIORXP3
AH8	VSS
AH9	RIORXP1
AH10	RIORXN1
AH11	VSS
AH12	PCIERXN1
AH13	PCIERXP1
AH14	VSS
AH15	CORESEL0
AH16	SRIOSGMIICLKN
AH17	PCIECLKP
AH18	PASSCLKN
AH19	GPIO07
AH19 †	BOOTMODE06
AH20	GPIO14
AH20 †	PCIESSMODE0
AH21	SPISCS0
AH22	SPIDIN
AH23	TIMO0
AH24	UARTCTS
AH25	EMU18
AH26	DVDD18
AH27	EMU16
AH28	EMU14
AH29	DVDD18
AJ1	VSS
AJ2	CVDD1
AJ3	VSS
AJ4	SGMII1RXN
AJ5	SGMII1RXP
AJ6	VSS
AJ7	RIORXN2
AJ8	RIORXP2
AJ9	VSS
AJ10	RIORXP0
AJ11	RIORXN0
AJ12	VSS
AJ13	PCIERXP0
AJ14	PCIERXN0
AJ15	VSS
AJ16	SRIOSGMIICLKP

Table 2-18 Terminal Functions — By Ball Number (Part 21 of 21)

Ball Number	Signal Name
AJ17	PCIECLKN
AJ18	PASSCLKP
AJ19	GPIO08
AJ19 †	BOOTMODE07
AJ20	GPIO00
AJ20 †	LENDIAN
AJ21	SPIDOUT
AJ22	SPISCS1
AJ23	TIMI0
AJ23 †	PCIESSEN
AJ24	UARTTXD
AJ25	RSV01
AJ26	EMU17
AJ27	EMU15
AJ28	DVDD18
AJ29	VSS
End of Table 2-18	

2.8 Development

2.8.1 Development Support

In case the customer would like to develop their own features and software on the C6670 device, TI offers an extensive line of development tools for the TMS320C6000™ DSP platform, including tools to evaluate the performance of the processors, generate code, develop algorithm implementations, and fully integrate and debug software and hardware modules. The tool's support documentation is electronically available within the Code Composer Studio™ Integrated Development Environment (IDE).

The following products support development of C6000™ DSP-based applications:

- **Software Development Tools:**
  - Code Composer Studio™ Integrated Development Environment (IDE), including Editor C/C++/Assembly Code Generation, and Debug plus additional development tools
  - Scalable, Real-Time Foundation Software (DSP/BIOS™), which provides the basic run-time target software needed to support any DSP application.
- **Hardware Development Tools:**
  - Extended Development System (XDS™) Emulator (supports C6000™ DSP multiprocessor system debug)
  - EVM (Evaluation Module)

2.8.2 Device Support

2.8.2.1 Device and Development-Support Tool Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all DSP devices and support tools. Each DSP commercial family member has one of three prefixes: TMX, TMP, or TMS (e.g., TMX320CMH). Texas Instruments recommends two of three possible prefix designators for its support tools: TMDX and TMDS. These prefixes represent evolutionary stages of product development from engineering prototypes (TMX/TMDX) through fully qualified production devices/tools (TMS/TMDS).

Device development evolutionary flow:

- **TMX:** Experimental device that is not necessarily representative of the final device's electrical specifications
- **TMP:** Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification
- **TMS:** Fully qualified production device

Support tool development evolutionary flow:

- **TMDX:** Development-support product that has not yet completed Texas Instruments internal qualification testing.
- **TMDS:** Fully qualified development-support product

TMX and TMP devices and TMDX development-support tools are shipped with the following disclaimer:

Developmental product is intended for internal evaluation purposes.

TMS devices and TMDS development-support tools have been characterized fully, and the quality and reliability of the device have been demonstrated fully. TI's standard warranty applies.

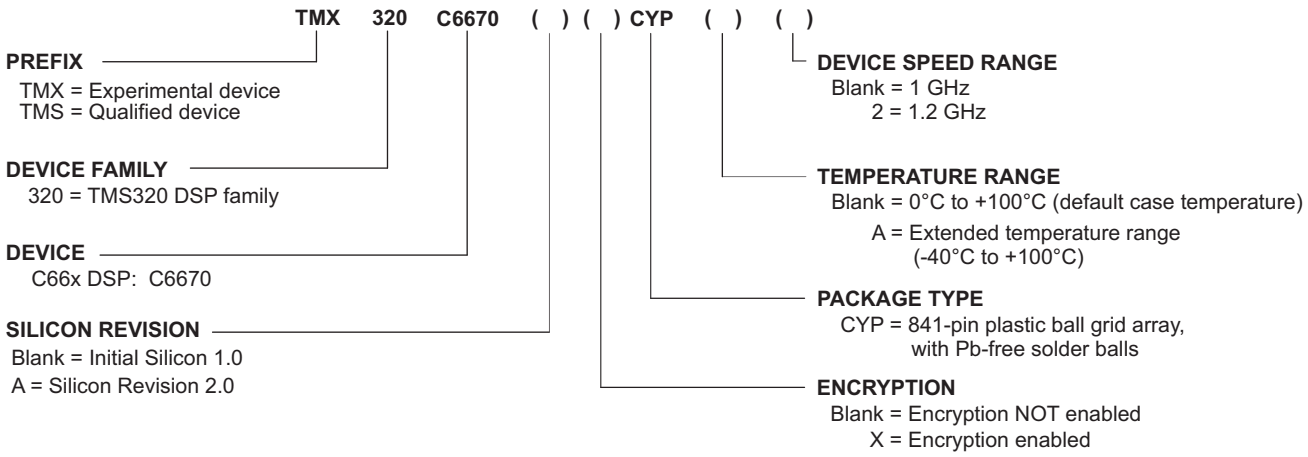
Predictions show that prototype devices (TMX or TMP) have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system because their expected end-use failure rate still is undefined. Only qualified production devices are to be used.

TI device nomenclature also includes a suffix with the device family name. This suffix indicates the package type (for example, CYP), the temperature range (for example, blank is the default case temperature range), and the device speed range, in Megahertz (for example, blank is 1000 MHz [1 GHz]).

For device part numbers and further ordering information for TMS320C6670 in the CYP package type, see the TI website www.ti.com or contact your TI sales representative.

Figure 2-17 provides a legend for reading the complete device name for any C66x+™ DSP generation member.

Figure 2-17 C66x™ DSP Device Nomenclature (including the TMS320C6670 DSP)



2.9 Related Documentation from Texas Instruments

These documents describe the TMS320C6670 Multicore Fixed and Floating-Point System-on-Chip. Copies of these documents are available on the Internet at www.ti.com

64-bit Timer (Timer 64) for KeyStone Devices User Guide	SPRUGV5
Antenna Interface 2 (AIF2) for KeyStone Devices User Guide	SPRUGV7
Bit Coprocessor (BCP) for KeyStone Devices User Guide	SPRUGZ1
Bootloader for the C66x DSP User Guide	SPRUGY5
C66x CorePac User Guide	SPRUGW0
C66x CPU and Instruction Set Reference Guide	SPRUGH7
C66x DSP Cache User Guide	SPRUGY8
DDR3 Design Guide for KeyStone Devices	SPRABI1
DSP Power Consumption Summary for KeyStone Devices	SPRABL4
Emulation and Trace Headers Technical Reference	SPRU655
Enhanced Direct Memory Access 3 (EDMA3) for KeyStone Devices User Guide	SPRUGS5
Fast Fourier Transform Coprocessor (FFTC) for KeyStone Devices User Guide	SPRUGS2
General Purpose Input/Output (GPIO) for KeyStone Devices User Guide	SPRUGV1
Gigabit Ethernet (GbE) Switch Subsystem for KeyStone Devices User Guide	SPRUGV9
Hardware Design Guide for KeyStone Devices	SPRABI2
HyperLink for KeyStone Devices User Guide	SPRUGW8
Inter Integrated Circuit (I^2C) for KeyStone Devices User Guide	SPRUGV3
Chip Interrupt Controller (CIC) for KeyStone Devices User Guide	SPRUGW4
Memory Protection Unit (MPU) for KeyStone Devices User Guide	SPRUGW5
Multicore Navigator for KeyStone Devices User Guide	SPRUGR9
Multicore Shared Memory Controller (MSMC) for KeyStone Devices User Guide	SPRUGW7
Network Coprocessor (NETCP) for KeyStone Devices User Guide	SPRUGZ6
Packet Accelerator (PA) for KeyStone Devices User Guide	SPRUGS4
Peripheral Component Interconnect Express (PCIe) for KeyStone Devices User Guide	SPRUGS6
Phase Locked Loop (PLL) Controller for KeyStone Devices User Guide	SPRUGV2
Power Sleep Controller (PSC) for KeyStone Devices User Guide	SPRUGV4
Serial Peripheral Interface (SPI) for KeyStone Devices User Guide	SPRUGP2
Serial RapidIO (SRIO) for KeyStone Devices User Guide	SPRUGW1
Turbo Decoder Coprocessor 3 (TCP3d) for KeyStone Devices User Guide	SPRUGS0
Turbo Encoder Coprocessor 3 (TCP3e) for KeyStone Devices User Guide	SPRUGS1
Universal Asynchronous Receiver/Transmitter (UART) for KeyStone Devices User Guide	SPRUGP1
Using Advanced Event Triggering to Debug Real-Time Problems in High Speed Embedded Microprocessor Systems	SPRA387
Using Advanced Event Triggering to Find and Fix Intermittent Real-Time Bugs	SPRA753
Using IBIS Models for Timing Analysis	SPRA839
Viterbi Coprocessor (VCP2) for KeyStone Devices User Guide	SPRUGV6

3 Device Configuration

On the TMS320C6670 device, certain device configurations like boot mode and endianess, are selected at device power-on reset. The status of the peripherals (enabled/disabled) is determined after device power-on reset. By default, the peripherals on the device are disabled and need to be enabled by software before being used.

3.1 Device Configuration at Device Reset

Table 3-1 describes the device configuration pins. The logic level is latched at power-on reset to determine the device configuration. The logic level on the device configuration pins can be set by using external pullup/pulldown resistors or by using some control device (e.g., FPGA/CPLD) to intelligently drive these pins. When using a control device, care should be taken to ensure there is no contention on the lines when the device is out of reset. The device configuration pins are sampled during power-on reset and are driven after the reset is removed. To avoid contention, the control device must stop driving the device configuration pins of the DSP.

Note—If a configuration pin must be routed out from the device and it is not driven (Hi-Z state), the internal pullup/pulldown (IPU/IPD) resistor should not be relied upon. TI recommends the use of an external pullup/pulldown resistor. For more detailed information on pullup/pulldown resistors and situations in which external pullup/pulldown resistors are required, see Section 3.4 "Pullup/Pulldown Resistors" on page 85.

Table 3-1 Device Configuration Pins

Configuration Pin	Pin No.	IPD/IPU (1)	Functional Description
LENDIAN(1) (2)	AJ20	IPU	Device endian mode (LENDIAN) 0 = Device operates in big endian mode 1 = Device operates in little endian mode
BOOTMODE[12:0] (1) (2)	AF21, AE20, AD20, AG19, AE21, AJ19, AH19, AG20, AE18, AF18, AE19, AD19, AG18	IPD	Method of boot See "Boot Modes Supported and PLL Settings" on page 30 for more details. See the Bootloader for the C66x DSP User Guide in "Related Documentation from Texas Instruments" on page 66 for detailed information on boot configuration
PCIESSMODE[1:0] (1) (2)	AD21, AH20	IPD	PCIe subsystem mode selection 00 = PCIe in end point mode 01 = PCIe legacy end point (support for legacy INTx) 10 = PCIe in root complex mode 11 = Reserved
PCIESSEN (1) (2)	AJ23	IPD	PCIe subsystem enable/disable 0 = PCIE subsystem is disabled 1 = PCIE subsystem is enabled
CORECLKSEL(1)	AB25	IPD	Core clock select 0 = SYSCLK is used as the input to Main PLL 1 = ALTCORECLK is used as the input to Main PLL
PACLKSEL(1)	AD23	IPD	Network coprocessor clock select 0 = SYSCLK / ALTCORECLK (controlled by CORECLKSEL pin) is used as the input to PA_SS PLL 1 = PASSCLK is used as the input to PASS PLL
End of Table 3-1			

1 Internal 100-µA pulldown or pullup is provided for this terminal. In most systems, a 1-kΩ resistor can be used to oppose the IPD/IPU. For more detailed information on pulldown/pullup resistors and situations in which external pulldown/pullup resistors are required, see Section 3.4 "Pullup/Pulldown Resistors" on page 85.

2 These signal names are the secondary functions of these pins.

3.2 Peripheral Selection After Device Reset

Several of the peripherals on the TMS320C6670 are controlled by the Power Sleep Controller (PSC). By default, the PCIe, SRIO, HyperLink, RAC, TAC, FFTC, AIF2, TCP3d, TCP3e, and VCP are held in reset and clock-gated. The memories in these modules are also in a low-leakage sleep mode. Software is required to turn these memories on. Then, the software enables the modules (turns on clocks and de-asserts reset) before these modules can be used.

If one of the above modules is used in the selected ROM boot mode, the ROM code will automatically enable the module.

All other modules come up enabled by default and there is no special software sequence to enable. For more detailed information on the PSC usage, see the *Power Sleep Controller (PSC) for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

3.3 Device State Control Registers

The TMS320C6670 device has a set of registers that are used to control the status of its peripherals. These registers are shown in Table 3-2.

Table 3-2 Device State Control Registers (Part 1 of 4)

Address Start	Address End	Size	Acronym	Description
0x02620000	0x02620007	8B	Reserved	
0x02620008	0x02620017	16B	Reserved	
0x02620018	0x0262001B	4B	JTAGID	See section 3.3.3
0x0262001C	0x0262001F	4B	Reserved	
0x02620020	0x02620023	4B	DEVSTAT	See section 3.3.1
0x02620024	0x02620037	20B	Reserved	
0x02620038	0x0262003B	4B	KICK0	See section 3.3.4
0x0262003C	0x0262003F	4B	KICK1	
0x02620040	0x02620043	4B	DSP_BOOT_ADDR0	The boot address for C66x DSP CorePac0
0x02620044	0x02620047	4B	DSP_BOOT_ADDR1	The boot address for C66x DSP CorePac1
0x02620048	0x0262004B	4B	DSP_BOOT_ADDR2	The boot address for C66x DSP CorePac2
0x0262004C	0x0262004F	4B	DSP_BOOT_ADDR3	The boot address for C66x DSP CorePac3
0x02620050	0x02620053	4B	Reserved	
0x02620054	0x02620057	4B	Reserved	
0x02620058	0x0262005B	4B	Reserved	
0x0262005C	0x0262005F	4B	Reserved	
0x02620060	0x026200DF	128B	Reserved	
0x026200E0	0x0262010F	48B	Reserved	
0x02620110	0x02620117	8B	MACID	See section 7.19 "Gigabit Ethernet (GbE) Switch Subsystem" on page 207
0x02620118	0x0262012F	24B	Reserved	
0x02620130	0x02620133	4B	LRSTNMIPINSTAT_CLR	See section 3.3.6
0x02620134	0x02620137	4B	RESET_STAT_CLR	See section 3.3.8
0x02620138	0x0262013B	4B	Reserved	
0x0262013C	0x0262013F	4B	BOOTCOMPLETE	See section 3.3.9
0x02620140	0x02620143	4B	Reserved	
0x02620144	0x02620147	4B	RESET_STAT	See section 3.3.7
0x02620148	0x0262014B	4B	LRSTNMIPINSTAT	See section 3.3.5
0x0262014C	0x0262014F	4B	DEVCFG	See section 3.3.2

Table 3-2 Device State Control Registers (Part 2 of 4)

Address Start	Address End	Size	Acronym	Description
0x02620150	0x02620153	4B	PWRSTATECTL	See section 3.3.10
0x02620154	0x02620157	4B	SRIO_SERDES_STS	See "Related Documentation from Texas Instruments" on page 66
0x02620158	0x0262015B	4B	SGMII_SERDES_STS	
0x0262015C	0x0262015F	4B	PCIE_SERDES_STS	
0x02620160	0x02620160	4B	HYPERLINK_SERDES_STS	
0x02620164	0x02620167	4B	AIF2_A_SERDES_STS	
0x02620168	0x0262016B	4B	AIF2_B_SERDES_STS	
0x0262016C	0x0262017F	20B	Reserved	
0x02620180	0x02620183	4B	SmartReflex Class0	
0x02620184	0x0262018F	12B	Reserved	
0x02620190	0x02620193	4B	Reserved	
0x02620194	0x02620197	4B	Reserved	
0x02620198	0x0262019B	4B	Reserved	
0x0262019C	0x0262019F	4B	Reserved	
0x026201A0	0x026201A3	4B	Reserved	
0x026201A4	0x026201A7	4B	Reserved	
0x026201A8	0x026201AB	4B	Reserved	
0x026201AC	0x026201AF	4B	Reserved	
0x026201B0	0x026201B3	4B	Reserved	
0x026201B4	0x026201B7	4B	Reserved	
0x026201B8	0x026201BB	4B	Reserved	
0x026201BC	0x026201BF	4B	Reserved	
0x026201C0	0x026201C3	4B	Reserved	
0x026201C4	0x026201C7	4B	Reserved	
0x026201C8	0x026201CB	4B	Reserved	
0x026201CC	0x026201CF	4B	Reserved	
0x026201D0	0x026201FF	48B	Reserved	
0x02620200	0x02620203	4B	NMIGR0	See section 3.3.11
0x02620204	0x02620207	4B	NMIGR1	
0x02620208	0x0262020B	4B	NMIGR2	
0x0262020C	0x0262020F	4B	NMIGR3	
0x02620210	0x02620213	4B	Reserved	
0x02620214	0x02620217	4B	Reserved	
0x02620218	0x0262021B	4B	Reserved	
0x0262021C	0x0262021F	4B	Reserved	
0x02620220	0x0262023F	32B	Reserved	
0x02620240	0x02620243	4B	IPCGR0	See section 3.3.12
0x02620244	0x02620247	4B	IPCGR1	
0x02620248	0x0262024B	4B	IPCGR2	
0x0262024C	0x0262024F	4B	IPCGR3	
0x02620250	0x02620253	4B	Reserved	
0x02620254	0x02620257	4B	Reserved	
0x02620258	0x0262025B	4B	Reserved	

Table 3-2 Device State Control Registers (Part 3 of 4)

Address Start	Address End	Size	Acronym	Description
0x0262025C	0x0262025F	4B	Reserved	
0x02620260	0x0262027B	28B	Reserved	
0x0262027C	0x0262027F	4B	IPCGRH	See section 3.3.14
0x02620280	0x02620283	4B	IPCAR0	See section 3.3.13
0x02620284	0x02620287	4B	IPCAR1	
0x02620288	0x0262028B	4B	IPCAR2	
0x0262028C	0x0262028F	4B	IPCAR3	
0x02620290	0x02620293	4B	Reserved	
0x02620294	0x02620297	4B	Reserved	
0x02620298	0x0262029B	4B	Reserved	
0x0262029C	0x0262029F	4B	Reserved	
0x026202A0	0x026202BB	28B	Reserved	
0x026202BC	0x026202BF	4B	IPCARH	See section 3.3.15
0x026202C0	0x026202FF	64B	Reserved	
0x02620300	0x02620303	4B	TINPSEL	See section 3.3.16
0x02620304	0x02620307	4B	TOUTPSEL	See section 3.3.17
0x02620308	0x0262030B	4B	RSTMUX0	See section 3.3.18
0x0262030C	0x0262030F	4B	RSTMUX1	
0x02620310	0x02620313	4B	RSTMUX2	
0x02620314	0x02620317	4B	RSTMUX3	
0x02620318	0x0262031B	4B	Reserved	
0x0262031C	0x0262031F	4B	Reserved	
0x02620320	0x02620323	4B	Reserved	
0x02620324	0x02620327	4B	Reserved	
0x02620328	0x0262032B	4B	MAINPLLCTL0	See section 7.5 "Main PLL and the PLL Controller" on page 128
0x0262032C	0x0262032F	4B	MAINPLLCTL1	
0x02620330	0x02620333	4B	DDR3PLLCTL0	See section 7.6 "DDR3 PLL" on page 142
0x02620334	0x02620337	4B	DDR3PLLCTL1	
0x02620338	0x0262033B	4B	PASSPLLCTL0	See section 7.7 "PASS PLL" on page 144
0x0262033C	0x0262033F	4B	PASSPLLCTL1	
0x02620340	0x02620343	4B	SGMII_SERDES_CFGPLL	See "Related Documentation from Texas Instruments" on page 66
0x02620344	0x02620347	4B	SGMII_SERDES_CFGRX0	
0x02620348	0x0262034B	4B	SGMII_SERDES_CFGTX0	
0x0262034C	0x0262034F	4B	SGMII_SERDES_CFGRX1	
0x02620350	0x02620353	4B	SGMII_SERDES_CFGTX1	
0x02620354	0x02620357	4B	Reserved	
0x02620358	0x0262035B	4B	PCIE_SERDES_CFGPLL	
0x0262035C	0x0262035F	4B	Reserved	
0x02620360	0x02620363	4B	SRIO_SERDES_CFGPLL	
0x02620364	0x02620367	4B	SRIO_SERDES_CFGRX0	
0x02620368	0x0262036B	4B	SRIO_SERDES_CFGTX0	
0x0262036C	0x0262036F	4B	SRIO_SERDES_CFGRX1	
0x02620370	0x02620373	4B	SRIO_SERDES_CFGTX1	

Table 3-2 Device State Control Registers (Part 4 of 4)

Address Start	Address End	Size	Acronym	Description
0x02620374	0x02620377	4B	SRIO_SERDES_CFGRX2	See "Related Documentation from Texas Instruments" on page 66
0x02620378	0x0262037B	4B	SRIO_SERDES_CFGTX2	
0x0262037C	0x0262037F	4B	SRIO_SERDES_CFGRX3	
0x02620380	0x02620383	4B	SRIO_SERDES_CFGTX3	
0x02620384	0x02620387	4B	Reserved	
0x02620388	0x026203AF	28B	Reserved	
0x026203B0	0x026203B3	4B	Reserved	
0x026203B4	0x026203B7	4B	HYPERLINK_SERDES_CFGPLL	See "Related Documentation from Texas Instruments" on page 66
0x026203B8	0x026203BB	4B	HYPERLINK_SERDES_CFGRX0	
0x026203BC	0x026203BF	4B	HYPERLINK_SERDES_CFGTX0	
0x026203C0	0x026203C3	4B	HYPERLINK_SERDES_CFGRX1	
0x026203C4	0x026203C7	4B	HYPERLINK_SERDES_CFGTX1	
0x026203C8	0x026203CB	4B	HYPERLINK_SERDES_CFGRX2	
0x026203CC	0x026203CF	4B	HYPERLINK_SERDES_CFGTX2	
0x026203D0	0x026203D3	4B	HYPERLINK_SERDES_CFGRX3	
0x026203D4	0x026203D7	4B	HYPERLINK_SERDES_CFGTX3	
0x026203D8	0x026203DB	4B	Reserved	
0x026203DC	0x026203F7	28B	Reserved	
0x026203F8	0x026203FB	4B	DEVSPEED	See section 3.3.19
0x026203FC	0x026203FF	4B	Reserved	
0x02620400	0x02620403	4B	PKTDMA_PRI_ALLOC	See section 4.4 "Bus Priorities" on page 97
0x02620404	0x02620467	100B	Reserved	
End of Table 3-2				

3.3.1 Device Status (DEVSTAT) Register

The Device Status Register depicts the device configuration selected upon a power-on reset by either the $\overline{\text{POR}}$ or $\overline{\text{RESETFULL}}$ pin. Once set, these bits will remain set until a power-on reset. The Device Status Register is shown in Figure 3-1 and described in Table 3-3.

Figure 3-1 Device Status Register

31 18	17	16	15 14	13 1	0
Reserved	PACLKSEL	PCIESSEN	PCIESSMODE[1:0	BOOTMODE[12:0]	LENDIAN
R-0		R-x	R/W-xx	R/W-xxxxxxxxxxxx	R-x (1)

Legend: R = Read only; RW = Read/Write; -*n* = value after reset

1 x indicates the bootstrap value latched via the external pin

Table 3-3 Device Status Register Field Descriptions

Bit	Field	Description
31-18	Reserved	Reserved. Read only, writes have no effect.
17	PACLKSEL	PA Clock select to select the reference clock for PA subsystem PLL 0 = Selects output of Main PLL MUX (SYSCLK vs. ALTCORECLK - depending on CORECLKSEL pin) 1 = Selects PASSCLKP/N
16	PCIESSEN	PCIe module enable 0 = PCIe module disabled 1 = PCIe module enabled
15-14	PCIESSMODE[1:0]	PCIe mode selection pins 00b = PCIe in end-point mode 01b = PCIe in legacy end-point mode (support for legacy INTx) 10b = PCIe in root complex mode 11b = Reserved
13-1	BOOTMODE[12:0]	Determines the bootmode configured for the device. For more information on bootmode, see Section 2.4 "Boot Modes Supported and PLL Settings" on page 30 and see the *Bootloader for the C66x DSP User Guide* in2.9 "Related Documentation from Texas Instruments" on page 66.
0	LENDIAN	Device endian mode (LENDIAN) — shows the status of whether the system is operating in big endian mode or little endian mode (default). 0 = System is operating in big endian mode 1 = System is operating in little endian mode (default)
End of Table 3-3		

3.3.2 Device Configuration Register

The Device Configuration Register is one-time writeable through software. The register is reset on all hard resets and is locked after the first write. The Device Configuration Register is shown in Figure 3-2 and described in Table 3-4.

Figure 3-2 Device Configuration Register (DEVCFG)

31	1	0
Reserved		SYSCLKOUTEN
R-0		R/W-1

Legend: R = Read only; RW = Read/Write; -n = value after reset

Table 3-4 Device Configuration Register Field Descriptions

Bit	Field	Description
31-1	Reserved	Reserved. Read only, writes have no effect.
0	SYSCLKOUTEN	SYSCLKOUT enable 0 = No clock output 1 = Clock output enabled (default)
End of Table 3-4		

3.3.3 JTAG ID (JTAGID) Register Description

The JTAG ID register is a read-only register that identifies to the customer the JTAG/Device ID. For the device, the JTAG ID register resides at address location 0x02620018. The JTAG ID Register is shown in the tables below.

Figure 3-3 JTAG ID (JTAGID) Register

31–28	27–12	11–1	0
VARIANT	PART NUMBER	MANUFACTURER	LSB
R-xxxx	R-1011 1001 0100 0001	0000 0010 111b	R-1

Legend: RW = Read/Write; R = Read only; -n = value after reset

Table 3-5 JTAG ID Register Field Descriptions

Bit	Field	Value	Description
31-28	VARIANT	xxxxb	Variant value
27-12	PART NUMBER	1011 1001 0100 0001b	Part Number for boundary scan
11-1	MANUFACTURER	0000 0010 111b	Manufacturer
0	LSB	1b	This bit is read as a 1 for TMS320C6670
End of Table 3-5			

Note—The value of the VARIANT and PART NUMBER fields depend on the silicon revision being used. See the Silicon Errata for details.

3.3.4 Kicker Mechanism (KICK0 and KICK1) Register

The Bootcfg module contains a kicker mechanism to prevent any spurious writes from changing any of the Bootcfg MMR values. When the kicker is locked (which it is initially after power on reset) none of the Bootcfg MMRs are writable (they are only readable). This mechanism requires two MMR writes to the KICK0 and KICK1 registers with exact data values before the kicker lock mechanism is un-locked. See Table 3-2 "Device State Control Registers" on page 68 for the address location. Once released then all the Bootcfg MMRs having write permissions are writable (the read only MMRs are still read only). The first KICK0 data is 0x83e70b13. The second KICK1 data is 0x95a4f1e0. Writing any other data value to either of these kick MMRs will lock the kicker mechanism and block any writes to Bootcfg MMRs. In order to ensure protection to all Bootcfg MMRs, software must always re-lock the kicker mechanism after completing the MMR writes.

3.3.5 LRESETNMI PIN Status (LRSTNMIPINSTAT) Register

The LRSTNMIPINSTAT Register is used to latch the status of $\overline{\text{LRESET}}$ and $\overline{\text{NMI}}$ based on the setting of CORESEL[2:0]. The LRESETNMI PIN Status Register is shown in Figure 3-4 and described in Table 3-6.

Figure 3-4 LRESETNMI PIN Status Register (LRSTNMIPINSTAT)

31–20	19	18	17	16	15–4	3	2	1	0
Reserved	NMI3	NMI2	NMI1	NMI0	Reserved	LR3	LR2	LR1	LR0
R, +000000000000	R-0	R-0	R-0	R-0	R, +000000000000	R-0	R-0	R-0	R-0

Legend: R = Read only; -*n* = value after reset

Table 3-6 LRESETNMI PIN Status Register Field Descriptions

Bit	Field	Description
31-20	Reserved	Reserved
19	NMI3	CorePac3 in NMI
18	NMI2	CorePac2 in NMI
17	NMI1	CorePac1 in NMI
16	NMI0	CorePac0 in NMI
15-4	Reserved	Reserved
3	LR4	CorePac3 in Local Reset
2	LR3	CorePac2 in Local Reset
1	LR31	CorePac1 in Local Reset
0	LR0	CorePac0 in Local Reset
End of Table 3-6		

3.3.6 LRESETNMI PIN Status Clear (LRSTNMIPINSTAT_CLR) Register

The LRSTNMIPINSTAT_CLR Register is used to clear the status of $\overline{\text{LRESET}}$ and $\overline{\text{NMI}}$ based on CORESEL[2:0]. The LRESETNMI PIN Status Clear Register is shown in Figure 3-5 and described in Table 3-7.

Figure 3-5 LRESETNMI PIN Status Clear Register (LRSTNMIPINSTAT_CLR)

31 20	19	18	17	16	15 4	3	2	1	0
Reserved	NMI3	NMI2	NMI1	NMI0	Reserved	LR3	LR2	LR1	LR0
R,+000000000000	WC,+0	WC,+0	WC,+0	WC,+0	R,+000000000000	WC,+0	WC,+0	WC,+0	WC,+0

Legend: R = Read only; -*n* = value after reset; WC = Write 1 to Clear

Table 3-7 LRESETNMI PIN Status Clear Register Field Descriptions

Bit	Field	Description
31-20	Reserved	Reserved
19	NMI3	CorePac3 in NMI Clear
18	NMI2	CorePac2 in NMI Clear
17	NMI1	CorePac1 in NMI Clear
16	NMI0	CorePac0 in NMI Clear
15-4	Reserved	Reserved
3	LR3	CorePac3 in Local Reset Clear
2	LR2	CorePac2 in Local Reset Clear
1	LR1	CorePac1 in Local Reset Clear
0	LR0	CorePac0 in Local Reset Clear
End of Table 3-7		

3.3.7 Reset Status (RESET_STAT) Register

The reset status register (RESET_STAT) captures the status of Local reset (LRx) for each of the cores and also the global device reset (GR). Software can use this information to take different device initialization steps, if desired.

- **In case of local reset:** The LRx bits are written as 1 and GR bit is written as 0 only when the CorePac receives an local reset without receiving a global reset.
- **In case of global reset:** The LRx bits are written as 0 and GR bit is written as 1 only when a global reset is asserted.

The Reset Status Register is shown in Figure 3-6 and described in Table 3-8.

Figure 3-6 Reset Status Register (RESET_STAT)

31	30 4	3	2	1	0
GR	Reserved	LR3	LR2	LR1	LR0
R, +1	R, + 000 0000 0000 0000 0000 0000 0000	R,+0	R,+0	R,+0	R,+0

Legend: R = Read only; -*n* = value after reset

Table 3-8 Reset Status Register Field Descriptions

Bit	Field	Description
31	GR	Global reset status 0 = Device has not received a global reset. 1 = Device received a global reset.
30-4	Reserved	Reserved.
3	LR3	CorePac3 reset status 0 = CorePac3 has not received a local reset. 1 = CorePac3 received a local reset.
2	LR2	CorePac2 reset status 0 = CorePac2 has not received a local reset. 1 = CorePac2 received a local reset.
1	LR1	CorePac1 reset status 0 = CorePac1 has not received a local reset. 1 = CorePac1 received a local reset.
0	LR0	CorePac0 reset status 0 = CorePac0 has not received a local reset. 1 = CorePac0 received a local reset.
End of Table 3-8		

3.3.8 Reset Status Clear (RESET_STAT_CLR) Register

The RESET_STAT bits can be cleared by writing 1 to the corresponding bit in the RESET_STAT_CLR register. The Reset Status Clear Register is shown in Figure 3-7 and described in Table 3-9.

Figure 3-7 Reset Status Clear Register (RESET_STAT_CLR)

31	30 4	3	2	1	0
GR	Reserved	LR3	LR2	LR1	LR0
RW, +0	R, + 000 0000 0000 0000 0000 0000 0000	RW,+0	RW,+0	RW,+0	RW,+0

Legend: R = Read only; RW = Read/Write; -*n* = value after reset

Table 3-9 Reset Status Clear Register Field Descriptions

Bit	Field	Description
31	GR	Global reset clear bit 0 = Writing a 0 has no effect. 1 = Writing a 1 to the GR bit clears the corresponding bit in the RESET_STAT register.
30-4	Reserved	Reserved.
3	LR3	CorePac3 reset clear bit 0 = Writing a 0 has no effect. 1 = Writing a 1 to the LR3 bit clears the corresponding bit in the RESET_STAT register.
2	LR2	CorePac2 reset clear bit 0 = Writing a 0 has no effect. 1 = Writing a 1 to the LR2 bit clears the corresponding bit in the RESET_STAT register.
1	LR1	CorePac1 reset clear bit 0 = Writing a 0 has no effect. 1 = Writing a 1 to the LR1 bit clears the corresponding bit in the RESET_STAT register.
0	LR0	CorePac0 reset clear bit 0 = Writing a 0 has no effect. 1 = Writing a 1 to the LR0 bit clears the corresponding bit in the RESET_STAT register.
End of Table 3-9		

3.3.9 Boot Complete (BOOTCOMPLETE) Register

The BOOTCOMPLETE register controls the BOOTCOMPLETE pin status. The purpose is to indicate the completion of the ROM booting process. The Boot Complete Register is shown in Figure 3-8 and described in Table 3-10.

Figure 3-8 Boot Complete Register (BOOTCOMPLETE)

31	4	3	2	1	0
Reserved		BC3	BC	BC1	BC0
R, + 0000 0000 0000 0000 0000 0000 0000		RW,+0	RW,+0	RW,+0	RW,+0

Legend: R = Read only; RW = Read/Write; *-n* = value after reset

Table 3-10 Boot Complete Register Field Descriptions

Bit	Field	Description
31-4	Reserved	Reserved.
3	BC3	CorePac 4 boot status 0 = CorePac 4 boot NOT complete 1 = CorePac 4 boot complete
2	BC2	CorePac3 boot status 0 = CorePac3 boot NOT complete 1 = CorePac3 boot complete
1	BC1	CorePac2 boot status 0 = CorePac2 boot NOT complete 1 = CorePac2 boot complete
0	BC0	CorePac1 boot status 0 = CorePac1 boot NOT complete 1 = CorePac1 boot complete
End of Table 3-10		

The BCx bit indicates the boot complete status of the corresponding CorePac. All BCx bits are sticky bits — that is they can be set only once by the software after device reset and they will be cleared to 0 on all device resets.

Boot ROM code will be implemented such that each CorePac will set its corresponding BCx bit immediately before branching to the predefined location in memory.

3.3.10 Power State Control (PWRSTATECTL) Register

The PWRSTATECTL register is controlled by the software to indicate the power-saving mode. ROM code reads this register to differentiate between the various power saving modes. This register is cleared only by POR and will survive all other device resets. See the *Hardware Design Guide for KeyStone Devices* in"Related Documentation from Texas Instruments" on page 66 for more information. The Power State Control Register is shown in Figure 3-9 and described in Table 3-11.

Figure 3-9 Power State Control Register (PWRSTATECTL)

31	3	2	1	0
GENERAL_PURPOSE		HIBERNATION_MODE	HIBERNATION	STANDBY
RW, +0000 0000 0000 0000 0000 0000 0000 0		RW,+0	RW,+0	RW,+0

Legend: RW = Read/Write; -*n* = value after reset

Table 3-11 Power State Control Register Field Descriptions

Bit	Field	Description
31-3	GENERAL_PURPOSE	Used to provide a start address for execution out of the hibernation modes. See the *Bootloader for the C66x DSP User Guide* in2.9 "Related Documentation from Texas Instruments" on page 66.
2	HIBERNATION_MODE	Indicates whether the device is in hibernation mode 1 or mode 2. 0 = Hibernation mode 1 1 = Hibernation mode 2
1	HIBERNATION	Indicates whether the device is in hibernation mode or not. 0 = Not in hibernation mode 1 = Hibernation mode
0	STANDBY	Indicates whether the device is in standby mode or not. 0 = Not in standby mode 1 = Standby mode
End of Table 3-11		

3.3.11 NMI Event Generation to CorePac (NMIGRx) Register

NMIGRx registers are used for generating NMI events to the corresponding CorePac. The C6670 has four NMIGRx registers (NMIGR0 through NMIGR3). The NMIGR0 register generates an NMI event to CorePac0, the NMIGR1 register generates an NMI event to CorePac1, and so on. Writing a 1 to the NMIG field generates a NMI pulse. Writing a 0 has no effect and Reads return 0 and have no other effect. The NMI Event Generation to CorePac Register is shown in Figure 3-10 and described in Table 3-12.

Figure 3-10 NMI Generation Register (NMIGRx)

31	1	0
Reserved		NMIG
R, +0000 0000 0000 0000 0000 0000 0000 000		RW,+0

Legend: RW = Read/Write; -*n* = value after reset

Table 3-12 NMI Generation Register Field Descriptions

Bit	Field	Description
31-1	Reserved	Reserved
0	NMIG	Reads return 0 Writes: 0 = No effect 1 = Creates NMI pulse to the corresponding CorePac — CorePac0 for NMIGR0, etc.
End of Table 3-12		

3.3.12 IPC Generation (IPCGRx) Registers

IPCGRx are the IPC interrupt generation registers to facilitate inter CorePac interrupts.

The C6670 has four IPCGRx registers (IPCGR0 through IPCGR3) registers. This can be used by external hosts or CorePacs to generate interrupts to other CorePacs. A write of 1 to IPCG field of IPCGRx register will generate an interrupt pulse to CorePacx (0 <= x <= 3).

These registers also provide a *Source ID* facility by which up to 28 different sources of interrupts can be identified. Allocation of source bits to source processor and meaning is entirely based on software convention. The register field descriptions are given in the following tables. Virtually anything can be a source for these registers as this is completely controlled by software. Any master that has access to BOOTCFG module space can write to these registers. The IPC Generation Register is shown in Figure 3-11 and described in Table 3-13.

Figure 3-11 IPC Generation Registers (IPCGRx)

31	30	29	28	27 – 8	7	6	5	4	3 – 1	0
SRCS27	SRCS26	SRCS25	SRCS24	SRCS23 – SRCS4	SRCS3	SRCS2	SRCS1	SRCS0	Reserved	IPCG
RW +0	RW +0	RW +0	RW +0	RW +0 (per bit field)	RW +0	RW +0	RW +0	RW +0	R, +000	RW +0

Legend: R = Read only; RW = Read/Write; -*n* = value after reset

Table 3-13 IPC Generation Registers Field Descriptions

Bit	Field	Description
31-4	SRCSx	Reads return current value of internal register bit. Writes: 0 = No effect 1 = Sets both SRCSx and the corresponding SRCCx.
3-1	Reserved	Reserved
0	IPCG	Reads return 0. Writes: 0 = No effect 1 = Creates an inter-DSP interrupt.
End of Table 3-13		

3.3.13 IPC Acknowledgement (IPCARx) Registers

IPCARx are the IPC interrupt-acknowledgement registers to facilitate inter-CorePac core interrupts.

The C6670 has four IPCARx (IPCAR0 through IPCAR3) registers. These registers also provide a *Source ID* facility by which up to 28 different sources of interrupts can be identified. Allocation of source bits to source processor and meaning is entirely based on software convention. The register field descriptions are given in the following tables. Virtually anything can be a source for these registers as this is completely controlled by software. Any master that has access to BOOTCFG module space can write to these registers. The IPC Acknowledgement Register is shown in Figure 3-12 and described in Table 3-14.

Figure 3-12 IPC Acknowledgement Registers (IPCARx)

31	30	29	28	27 – 8	7	6	5	4	3 – 0
SRCC27	SRCC26	SRCC25	SRCC24	SRCC23 – SRCC4	SRCC3	SRCC2	SRCC1	SRCC0	Reserved
RW +0	RW +0	RW +0	RW +0	RW +0 (per bit field)	RW +0	RW +0	RW +0	RW +0	R, +0000

Legend: R = Read only; RW = Read/Write; -*n* = value after reset

Table 3-14 IPC Acknowledgement Registers Field Descriptions

Bit	Field	Description
31-4	SRCCx	Reads return current value of internal register bit. Writes: 0 = No effect 1 = Clears both SRCCx and the corresponding SRCSx
3-0	Reserved	Reserved
End of Table 3-14		

3.3.14 IPC Generation Host (IPCGRH) Register

IPCGRH register is provided to facilitate host CPU interrupt. Operation and use of IPCGRH is the same as other IPCGR registers. Interrupt output pulse created by IPCGRH is driven on a device pin, host interrupt/event output (HOUT).

The host interrupt output pulse should be stretched. It should be asserted for 4 bootcfg clock cycles (CPU/6) followed by a deassertion of 4 bootcfg clock cycles. Generating the pulse will result in 8 CPU/6 cycle pulse blocking window. Write to IPCGRH with IPCG bit (bit 0) set will only generate a pulse if they are beyond 8 CPU/6 cycle period. The IPC Generation Host Register is shown in Figure 3-13 and described in Table 3-15.

Figure 3-13 IPC Generation Registers (IPCGRH)

31	30	29	28	27 – 8	7	6	5	4	3 – 1	0
SRCS27	SRCS26	SRCS25	SRCS24	SRCS23 – SRCS4	SRCS3	SRCS2	SRCS1	SRCS0	Reserved	IPCG
RW +0	RW +0	RW +0	RW +0	RW +0 (per bit field)	RW +0	RW +0	RW +0	RW +0	R, +000	RW +0

Legend: R = Read only; RW = Read/Write; -*n* = value after reset

Table 3-15 IPC Generation Registers Field Descriptions

Bit	Field	Description
31-4	SRCSx	Reads return current value of internal register bit. Writes: 0 = No effect 1 = Sets both SRCSx and the corresponding SRCCx.
3-1	Reserved	Reserved
0	IPCG	Reads return 0. Writes: 0 = No effect 1 = Creates an interrupt pulse on device pin (host interrupt/event output in HOUT pin)
End of Table 3-15		

3.3.15 IPC Acknowledgement Host (IPCARH) Register

IPCARH registers are provided to facilitate host CPU interrupt. Operation and use of IPCARH is the same as other IPCAR registers. The IPC Acknowledgement Host Register is shown in Figure 3-14 and described in Table 3-16.

Figure 3-14 IPC Acknowledgement Register (IPCARH)

31	30	29	28	27 – 8	7	6	5	4	3 – 0
SRCC27	SRCC26	SRCC25	SRCC24	SRCC23 – SRCC4	SRCC3	SRCC2	SRCC1	SRCC0	Reserved
RW +0	RW +0	RW +0	RW +0	RW +0 (per bit field)	RW +0	RW +0	RW +0	RW +0	R, +0000

Legend: R = Read only; RW = Read/Write; *-n* = value after reset

Table 3-16 IPC Acknowledgement Register Field Descriptions

Bit	Field	Description
31-4	SRCCx	Reads return current value of internal register bit. Writes: 0 = No effect 1 = Clears both SRCCx and the corresponding SRCSx
3-0	Reserved	Reserved
End of Table 3-16		

3.3.16 Timer Input Selection Register (TINPSEL)

Timer input selection is handled within the control register TINPSEL. The Timer Input Selection Register is shown in Figure 3-15 and described in Table 3-17.

Figure 3-15 Timer Input Selection Register (TINPSEL)

31 – 16	15	14	13	12	11	10	9
Reserved	TINPHSEL7	TINPLSEL7	TINPHSEL6	TINPLSEL6	TINPHSEL5	TINPLSEL5	TINPHSEL4
0	RW, +1	RW, +0	RW, +1	RW, +0	RW, +1	RW, +0	RW, +1

8	7	6	5	4	3	2	1	0
TINPLSEL4	TINPHSEL3	TINPLSEL3	TINPHSEL2	TINPLSEL2	TINPHSEL1	TINPLSEL1	TINPHSEL0	TINPLSEL0
RW, +0	RW, +1	RW, +0	RW, +1	RW, +0	RW, +1	RW, +1	RW, +1	RW, +0

Legend: R = Read only; RW = Read/Write; *-n* = value after reset

Table 3-17 Timer Input Selection Field Description

Bit	Field	Description
31-16	Reserved	Reserved
15	TINPHSEL7	Input select for TIMER7 high. 0 = TIMI0 1 = TIMI1
14	TINPLSEL7	Input select for TIMER7 low. 0 = TIMI0 1 = TIMI1
13	TINPHSEL6	Input select for TIMER6 high. 0 = TIMI0 1 = TIMI1
12	TINPLSEL6	Input select for TIMER6 low. 0 = TIMI0 1 = TIMI1
11	TINPHSEL5	Input select for TIMER5 high. 0 = TIMI0 1 = TIMI1
10	TINPLSEL5	Input select for TIMER5 low. 0 = TIMI0 1 = TIMI1
9	TINPHSEL4	Input select for TIMER4 high. 0 = TIMI0 1 = TIMI1
8	TINPLSEL4	Input select for TIMER4 low. 0 = TIMI0 1 = TIMI1
7	TINPHSEL3	Input select for TIMER3 high. 0 = TIMI0 1 = TIMI1
6	TINPLSEL3	Input select for TIMER3 low. 0 = TIMI0 1 = TIMI1
5	TINPHSEL2	Input select for TIMER2 high. 0 = TIMI0 1 = TIMI1
4	TINPLSEL2	Input select for TIMER2 low. 0 = TIMI0 1 = TIMI1
3	TINPHSEL1	Input select for TIMER1 high. 0 = TIMI0 1 = TIMI1
2	TINPLSEL1	Input select for TIMER1 low. 0 = TIMI0 1 = TIMI1
1	TINPHSEL0	Input select for TIMER0 high. 0 = TIMI0 1 = TIMI1
0	TINPLSEL0	Input select for TIMER0 low. 0 = TIMI0 1 = TIMI1
End of Table 3-17		

3.3.17 Timer Output Selection Register (TOUTPSEL)

The timer output selection is handled within the control register TOUTSEL. The Timer Output Selection Register is shown in Figure 3-16 and described in Table 3-18.

Figure 3-16 Timer Output Selection Register (TOUTPSEL)

31 9	8 5	4	3 0
Reserved	TOUTPSEL1	Reserved	TOUTPSEL0
R,+00000000000000000000000000	RW,+0001	0	RW,+0000

Legend: R = Read only; RW = Read/Write; -*n* = value after reset

Table 3-18 Timer Output Selection Field Description

Bit	Field	Description
31-9	Reserved	Reserved
8-5	TOUTPSEL1	Output select for TIMO1 0000: TOUTL0 0001: TOUTH0 0010: TOUTL1 0011: TOUTH1 0100: TOUTL2 0101: TOUTH2 0110: TOUTL3 0111: TOUTH3 1000: TOUTL4 1001: TOUTH4 1010: TOUTL5 1011: TOUTH5 1100: TOUTL6 1101: TOUTH6 1110: TOUTL7 1111: TOUTH7
4	Reserved	Reserved
3-0	TOUTPSEL0	Output select for TIMO0 0000: TOUTL0 0001: TOUTH0 0010: TOUTL1 0011: TOUTH1 0100: TOUTL2 0101: TOUTH2 0110: TOUTL3 0111: TOUTH3 1000: TOUTL4 1001: TOUTH4 1010: TOUTL5 1011: TOUTH5 1100: TOUTL6 1101: TOUTH6 1110: TOUTL7 1111: TOUTH7
End of Table 3-18		

3.3.18 Reset Mux (RSTMUXx) Register

The software controls the Reset Mux block through the reset multiplex registers using RSTMUX0 through RSTMUX3 for each of the four CorePacs on the C6670. These registers are located in Bootcfg memory space. The Timer Output Selection Register is shown in Figure 3-17 and described in Table 3-19.

Figure 3-17 Reset Mux Register (RSTMUX0 through RSTMUX3)

31 10	9	8	7 5	4	3 1	0
Reserved	EVTSTATCLR	Reserved	DELAY	EVTSTAT	OMODE	LOCK
R, +0000 0000 0000 0000 0000 00	RC, +0	R, +0	RW, +100	R, +0	RW, +000	RW, +0

Legend: R = Read only; RW = Read/Write; -*n* = value after reset; RC = Read only and write 1 to clear

Table 3-19 Reset Mux Register Field Descriptions

Bit	Field	Description
31-10	Reserved	Reserved
9	EVTSTATCLR	Clear event status 0 = Writing 0 has no effect 1 = Writing 1 to this bit clears the EVTSTAT bit
8	Reserved	Reserved
7-5	DELAY	Delay cycles between NMI & local reset 000b = 256 CPU/6 cycles delay between NMI & local reset, when OMODE = 100b 001b = 512 CPU/6 cycles delay between NMI & local reset, when OMODE=100b 010b = 1024 CPU/6 cycles delay between NMI & local reset, when OMODE=100b 011b = 2048 CPU/6 cycles delay between NMI & local reset, when OMODE=100b 100b = 4096 CPU/6 cycles delay between NMI & local reset, when OMODE=100b (default) 101b = 8192 CPU/6 cycles delay between NMI & local reset, when OMODE=100b 110b = 16384 CPU/6 cycles delay between NMI & local reset, when OMODE=100b 111b = 32768 CPU/6 cycles delay between NMI & local reset, when OMODE=100b
4	EVTSTAT	Event status 0 = No event received (Default) 1 = WD timer event received by Reset Mux block
3-1	OMODE	Timer event operation mode 000b = WD timer event input to the Reset Mux block does not cause any output event (default) 001b = Reserved 010b = WD Timer Event input to the Reset Mux block causes local reset input to CorePac 011b = WD Timer Event input to the Reset Mux block causes NMI input to CorePac 100b = WD Timer Event input to the Reset Mux block causes NMI input followed by local reset input to CorePac. Delay between NMI and local reset is set in DELAY bit field. 101b = WD timer event input to the Reset Mux block causes device reset to C6670 110b = Reserved 111b = Reserved
0	LOCK	Lock register fields 0 = Register fields are not locked (default) 1 = Register fields are locked until the next timer reset
End of Table 3-19		

3.3.19 Device Speed (DEVSPEED) Register

The Device Speed Register shows the device speed grade. The Device Speed Register is shown below.

Figure 3-18 Device Speed Register (DEVSPEED)

31	23	22	0
DEVSPEED		Reserved	
R-n		R-n	

Legend: R = Read only; -*n* = value after reset

Table 3-20 Device Speed Register Field Descriptions

Bit	Field	Description
31-23	DEVSPEED	Indicates the speed of the device (read only) 0b0000 0000 0 = 800 MHz 0b0000 0000 1 = 1000 MHz 0b0000 0001 x = 1200 MHz 0b001x xxxx x = 1200 MHz 0b01xx xxxx x = 1000 MHz 0b1xxx xxxx x = 800 MHz
22-0	Reserved	Reserved. Read only
End of Table 3-20		

3.4 Pullup/Pulldown Resistors

Proper board design should ensure that input pins to the device always be at a valid logic level and not floating. This may be achieved via pullup/pulldown resistors. The device features internal pullup (IPU) and internal pulldown (IPD) resistors on most pins to eliminate the need, unless otherwise noted, for external pullup/pulldown resistors.

An external pullup/pulldown resistor needs to be used in the following situations:

- **Device Configuration Pins:** If the pin is both routed out and are not driven (in Hi-Z state), an external pullup/pulldown resistor must be used, even if the IPU/IPD matches the desired value/state.
- **Other Input Pins:** If the IPU/IPD does not match the desired value/state, use an external pullup/pulldown resistor to pull the signal to the opposite rail.

For the device configuration pins (listed in Table 3-1), if they are both routed out and are not driven (in Hi-Z state), it is strongly recommended that an external pullup/pulldown resistor be implemented. Although, internal pullup/pulldown resistors exist on these pins and they may match the desired configuration value, providing external connectivity can help ensure that valid logic levels are latched on these device configuration pins. In addition, applying external pullup/pulldown resistors on the device configuration pins adds convenience to the user in debugging and flexibility in switching operating modes.

Tips for choosing an external pullup/pulldown resistor:

- Consider the total amount of current that may pass through the pullup or pulldown resistor. Make sure to include the leakage currents of all the devices connected to the net, as well as any internal pullup or pulldown resistors.
- Decide a target value for the net. For a pulldown resistor, this should be below the lowest V_{IL} level of all inputs connected to the net. For a pullup resistor, this should be above the highest V_{IH} level of all inputs on the net. A reasonable choice would be to target the V_{OL} or V_{OH} levels for the logic family of the limiting device; which, by definition, have margin to the V_{IL} and V_{IH} levels.
- Select a pullup/pulldown resistor with the largest possible value that can still ensure that the net will reach the target pulled value when maximum current from all devices on the net is flowing through the resistor. The current to be considered includes leakage current plus, any other internal and external pullup/pulldown resistors on the net.
- For bidirectional nets, there is an additional consideration that sets a lower limit on the resistance value of the external resistor. Verify that the resistance is small enough that the weakest output buffer can drive the net to the opposite logic level (including margin).
- Remember to include tolerances when selecting the resistor value.
- For pullup resistors, also remember to include tolerances on the DVDD rail.

For most systems:

- A 1-kΩ resistor can be used to oppose the IPU/IPD while meeting the above criteria. Users should confirm this resistor value is correct for their specific application.
- A 20-kΩ resistor can be used to compliment the IPU/IPD on the device configuration pins while meeting the above criteria. Users should confirm this resistor value is correct for their specific application.

For more detailed information on input current (I_I), and the low-level/high-level input voltages (V_{IL} and V_{IH}) for the TMS320C6670 device, see Section 6.3 "Electrical Characteristics" on page 107.

To determine which pins on the device include internal pullup/pulldown resistors, see Table 2-16 "Terminal Functions — Power and Ground" on page 52.

4 System Interconnect

On the TMS320C6670 device, the C66x CorePacs, the EDMA3 transfer controllers, and the system peripherals are interconnected through the TeraNet, which is a non-blocking switch fabric enabling fast and contention-free internal data movement. The TeraNet provides low-latency, concurrent data transfers between master peripherals and slave peripherals. The TeraNet also allows for seamless arbitration between the system masters when accessing system slaves.

4.1 Internal Buses and Switch Fabrics

Two types of buses exist in the device; data buses and configuration buses. Some peripherals have both a data bus and a configuration bus interface, while others only have one type of interface. Furthermore, the bus interface width and speed varies from peripheral to peripheral. Configuration buses are mainly used to access the register space of a peripheral and the data buses are used mainly for data transfers.

The C66x CorePacs, the EDMA3 traffic controllers, and the various system peripherals can be classified into two categories: masters and slaves. Masters are capable of initiating read and write transfers in the system and do not rely on the EDMA3 for their data transfers. Slaves on the other hand rely on the masters to perform transfers to and from them. Examples of masters include the EDMA3 traffic controllers, SRIO, and network coprocessor packet DMA. Examples of slaves include the SPI, UART, and I^2C.

The masters and slaves in the device are communicating through the TeraNet (switch fabric). The device contains two switch fabrics. The data switch fabric (data TeraNet) and the configuration switch fabric (configuration TeraNet). The data TeraNet, is a high-throughput interconnect mainly used to move data across the system. The data TeraNet connects masters to slaves via data buses. The configuration TeraNet, is mainly used to access peripheral registers. The configuration TeraNet connects masters to slaves via configuration buses. Note that the data TeraNet also connects to the configuration TeraNet. For more details see 4.2 "Switch Fabric Connections Matrix" on page 87.

4.2 Switch Fabric Connections Matrix

The tables below list the master and slave end point connections.

Intersecting cells may contain one of the following:

- **Y** — There is a connection between this master and that slave.
- **-** — There is NO connection between this master and that slave.
- ***n*** — A numeric value indicates that the path between this master and that slave goes through bridge *n*.

Table 4-1 Switch Fabric Connection Matrix Section 1 (Part 1 of 2)

	Slave																							
Masters	**CorePac0_SDMA**	**CorePac1_SDMA**	**CorePac2_SDMA**	**CorePac3_SDMA**	**Boot_ROM**	**SPI**	**PCIe_Slave**	**QM_Slave**	**HyperLink_Slave**	**MSMC_SES**	**MSMC_SMS**	**STM**	**TETB_System**	**TETB0**	**TETB1**	**TETB2**	**TETB3**	**VCP2(0-4)**	**TCP3d**	**TCP_3e_w**	**TCP3e_r**	**TAC_BE**	**RAC_Slave**	**BCP_FFTCC_TCP3dC_S**
HyperLink_Master	1	1	1	1	1	1	1	1	-	Y	Y	-	-	-	-	-	-	Y	Y	Y	Y	-	-	-
BCP_FFTCC_TCP3dC Master	Y	Y	Y	Y	-	-	-	Y	-	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC0_TC0_RD	2	2	2	2	2	2	2	2	Y	Y	Y	-	Y	-	-	-	-	-	-	-	-	-	-	Y
EDMA3CC0_TC0_WR	2	2	2	2	-	2	2	2	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-	Y
EDMA3CC0_TC1_RD	3	3	3	3	3	3	3	-	Y	Y	Y	-	Y	-	-	-	-	-	-	-	-	-	-	Y
EDMA3CC0_TC1_WR	3	3	3	3	-	3	3	-	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-	Y
EDMA3CC1_TC0_RD	Y	Y	Y	Y	Y	Y	Y	-	5	5	5	-	Y	-	-	-	-	-	-	-	-	-	Y	-
EDMA3CC1_TC0_WR	Y	Y	Y	Y	-	Y	Y	-	5	5	5	Y	-	-	-	-	-	-	-	-	-	-	Y	-
EDMA3CC1_TC1_RD	Y	Y	Y	Y	Y	Y	Y	Y	6	6	6	-	-	Y	Y	-	-	-	-	-	-	-	Y	-
EDMA3CC1_TC1_WR	Y	Y	Y	Y	-	Y	Y	Y	6	6	6	-	-	-	-	-	-	-	-	-	-	-	Y	-
EDMA3CC1_TC2_RD	Y	Y	Y	Y	Y	Y	Y	-	7	7	7	-	-	-	-	Y	Y	-	-	-	-	Y	-	-
EDMA3CC1_TC2_WR	Y	Y	Y	Y	-	Y	Y	-	7	7	7	-	-	-	-	-	-	-	-	-	-	Y	-	-
EDMA3CC1_TC3_RD	Y	Y	Y	Y	Y	Y	Y	-	8	8	8	-	Y	-	-	-	-	-	-	-	-	Y	-	-
EDMA3CC1_TC3_WR	Y	Y	Y	Y	-	Y	Y	-	8	8	8	Y	-	-	-	-	-	-	-	-	-	Y	-	-
EDMA3CC2_TC0_RD	Y	Y	Y	Y	Y	Y	Y	-	9	9	9	-	Y	-	-	-	-	Y	Y	Y	Y	-	-	Y
EDMA3CC2_TC0_WR	Y	Y	Y	Y	-	Y	Y	-	9	9	9	Y	-	-	-	-	-	Y	Y	Y	Y	-	-	Y
EDMA3CC2_TC1_RD	Y	Y	Y	Y	Y	Y	Y	Y	10	10	10	-	-	Y	Y	-	-	Y	Y	Y	Y	-	-	Y
EDMA3CC2_TC1_WR	Y	Y	Y	Y	-	Y	Y	Y	10	10	10	-	-	-	-	-	-	Y	Y	Y	Y	-	-	Y
EDMA3CC2_TC2_RD	Y	Y	Y	Y	Y	Y	Y	-	5	5	5	-	Y	-	-	-	-	Y	Y	-	-	-	-	Y
EDMA3CC2_TC2_WR	Y	Y	Y	Y	-	Y	Y	-	5	5	5	Y	-	-	-	-	-	Y	Y	-	-	-	-	Y
EDMA3CC2_TC3_RD	Y	Y	Y	Y	Y	Y	Y	-	6	6	6	-	-	-	-	Y	Y	Y	-	Y	Y	-	-	Y
EDMA3CC2_TC3_WR	Y	Y	Y	Y	-	Y	Y	-	6	6	6	-	-	-	-	-	-	Y	-	Y	Y	-	-	Y
SRIO Packet DMA	Y	Y	Y	Y	-	-	-	Y	-	9	9	-	-	-	-	-	-	-	-	-	-	-	-	-
SRIO_Master	Y	Y	Y	Y	-	Y	-	Y	7	7	7	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
PCIe_Master	Y	Y	Y	Y	-	Y	-	Y	7	7	7	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Network Coprocessor Packet DMA	Y	Y	Y	Y	-	-	-	Y	-	10	10	-	-	-	-	-	-	-	-	-	-	-	-	-
MSMC_Data_Master	4	4	4	4	4	4	4	4	Y	-	-	Y	-	-	-	-	-	Y	Y	Y	Y	Y	Y	Y
QM_SS_Master	Y	Y	Y	Y	-	-	-	Y	8	8	8	-	-	-	-	-	-	-	-	-	-	-	-	-
QM_SS_Second	Y	Y	Y	Y	-	-	-	-	8	8	8	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 4-1 Switch Fabric Connection Matrix Section 1 (Part 2 of 2)

Masters	Slave																						
	CorePac0_SDMA	CorePac1_SDMA	CorePac2_SDMA	CorePac3_SDMA	Boot_ROM	SPI	PCIe_Slave	QM_Slave	HyperLink_Slave	MSMC_SES	MSMC_SMS	STM	TETB_System	TETB0	TETB1	TETB2	TETB3	VCP2(0-4)	TCP3d	TCP_3e_w	TCP3e_r	TAC_BE	RAC_Slave	BCP_FFTCC_TCP3dC_S
DebugSS_Master	Y	Y	Y	Y	Y	Y	Y	Y	10	10	10	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
FFTC	Y	Y	Y	Y	-	-	-	Y	6	6	6	-	-	-	-	-	-	-	-	-	-	-	-	-
RAC_BE0	Y	Y	Y	Y	-	-	-	-	7	7	7	-	-	-	-	-	-	-	-	-	-	-	-	-
RAC_BE1	Y	Y	Y	Y	-	-	-	-	8	8	8	-	-	-	-	-	-	-	-	-	-	-	-	-
AIF_Master	Y	Y	Y	Y	-	-	-	Y	7	7	7	-	-	-	-	-	-	-	-	-	-	Y	Y	-
TAC_FE	Y	Y	Y	Y	-	-	-	-	9	9	9	-	-	-	-	-	-	-	-	-	-	-	-	-
CorePac0_CFG	-	-	-	-	-	-	-	-	-	-	-	-	Y	Y	Y	Y	Y	-	-	-	-	-	-	-
CorePac1_CFG	-	-	-	-	-	-	-	-	-	-	-	-	Y	Y	Y	Y	Y	-	-	-	-	-	-	-
CorePac2_CFG	-	-	-	-	-	-	-	-	-	-	-	-	Y	Y	Y	Y	Y	-	-	-	-	-	-	-
CorePac3_CFG	-	-	-	-	-	-	-	-	-	-	-	-	Y	Y	Y	Y	Y	-	-	-	-	-	-	-
Tracer_Master	-	-	-	-	-	-	-	-	-	-	-	Y	-	-	-	-	-	-	-	-	-	-	-	-
End of Table 4-1																								

Table 4-2 Switch Fabric Connection Matrix Section 2 (Part 1 of 2)

Masters	Slave																						
	EDMA3CC0	EDMA3CC1	EDMA3CC2	EDMA3CC0_TC(0-1)	EDMA3CC1_TC(0-3)	EDMA3CC2_TC(0-3)	Semaphore	QM_SS_CFG	CP Tracer(0~15/16)	NETCP_CFG	SRIO_CFG	Timer	GPIO	I^2C	SEC_CTL	SEC_Key_MGR	Boot_CFG	GPSC	PLL_CTL	CP_CIC	MPU	Debug_SS_CFG	SR_MMR
HyperLink_Master	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12
BCP_FFTCC_TCP3dC Master	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	-	-	-	-	-	Y	Y	Y	-
EDMA3CC0_TC0_RD	2, 12	2, 12	2, 12	2, 12	2, 12	2, 12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC0_TC0_WR	2, 12	2, 12	2, 12	2, 12	2, 12	2, 12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC0_TC1_RD	3, 12	3, 12	3, 12	3, 12	3, 12	3, 12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC0_TC1_WR	3, 12	3, 12	3, 12	3, 12	3, 12	3, 12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC1_TC0_RD	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA3CC1_TC0_WR	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA3CC1_TC1_RD	13	13	13	13	13	13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC1_TC1_WR	13	13	13	13	13	13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC1_TC2_RD	14	14	14	14	14	14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC1_TC2_WR	14	14	14	14	14	14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC1_TC3_RD	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12

Table 4-2 Switch Fabric Connection Matrix Section 2 (Part 2 of 2)

Masters	Slave																						
	EDMA3CC0	EDMA3CC1	EDMA3CC2	EDMA3CC0_TC(0-1)	EDMA3CC1_TC(0-3)	EDMA3CC2_TC(0-3)	Semaphore	QM_SS_CFG	CP Tracer(0~15/16)	NETCP_CFG	SRIO_CFG	Timer	GPIO	I^2C	SEC_CTL	SEC_Key_MGR	Boot_CFG	GPSC	PLL_CTL	CP_CIC	MPU	Debug_SS_CFG	SR_MMR
EDMA3CC1_TC3_WR	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA3CC2_TC0_RD	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA3CC2_TC0_WR	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA3CC2_TC1_RD	13	13	13	13	13	13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC2_TC1_WR	13	13	13	13	13	13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC2_TC2_RD	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA3CC2_TC2_WR	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA3CC2_TC3_RD	14	14	14	14	14	14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC2_TC3_WR	14	14	14	14	14	14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRIO Packet DMA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRIO_M	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
PCIe_Master	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
Network Coprocessor Packet DMA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MSMC_Data_Master	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
QM_SS Packet DMA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
QM_SS Second	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DebugSS_Master	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
FFTC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RAC_BE0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RAC_BE1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
AIF_Master	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TAC_FE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC0	-	-	-	Y	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC1	-	-	-	-	Y	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC2	-	-	-	-	-	Y	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CorePac0_CFG	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
CorePac1_CFG	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
CorePac2_CFG	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
CorePac3_CFG	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
End of Table 4-2																							

Table 4-3 Switch Fabric Connection Matrix Section 3 (Part 1 of 2)

Masters	Slave							
	RAC_CFG	FFTC_CFG	TAC_CFG	TCP3e_CFG	TCP3d_CFG	VCP2_CFG	AIF2_CFG	UART_CFG
HyperLink_Master	1, 12	Y	1, 12	1, 12	1, 12	1, 12	Y	Y
BCP_FFTCC_TCP3dC Master	Y	Y	Y	Y	Y	Y	Y	Y
EDMA3CC0_TC0_RD	-	-	-	-	-	-	-	-

Table 4-3 Switch Fabric Connection Matrix Section 3 (Part 2 of 2)

Masters	Slave							
	RAC_CFG	FFTC_CFG	TAC_CFG	TCP3e_CFG	TCP3d_CFG	VCP2_CFG	AIF2_CFG	UART_CFG
EDMA3CC0_TC0_WR	-	-	-	-	-	-	-	-
EDMA3CC0_TC1_RD	-	-	-	-	-	-	-	-
EDMA3CC0_TC1_WR	-	-	-	-	-	-	-	-
EDMA3CC1_TC0_RD	12	Y	12	12	12	12	Y	Y
EDMA3CC1_TC0_WR	12	Y	12	12	12	12	Y	Y
EDMA3CC1_TC1_RD	-	-	-	-	-	-	-	-
EDMA3CC1_TC1_WR	-	-	-	-	-	-	-	-
EDMA3CC1_TC2_RD	-	-	-	-	-	-	-	-
EDMA3CC1_TC2_WR	-	-	-	-	-	-	-	-
EDMA3CC1_TC3_RD	12	Y	12	12	12	12	Y	Y
EDMA3CC1_TC3_WR	12	Y	12	12	12	12	Y	Y
EDMA3CC2_TC0_RD	12	Y	12	12	12	12	Y	Y
EDMA3CC2_TC0_WR	12	Y	12	12	12	12	Y	Y
EDMA3CC2_TC1_RD	-	-	-	-	-	-	-	-
EDMA3CC2_TC1_WR	-	-	-	-	-	-	-	-
EDMA3CC2_TC2_RD	12	Y	12	12	12	12	Y	Y
EDMA3CC2_TC2_WR	12	Y	12	12	12	12	Y	Y
EDMA3CC2_TC3_RD	-	-	-	-	-	-	-	-
EDMA3CC2_TC3_WR	-	-	-	-	-	-	-	-
SRIO Packet DMA	-	-	-	-	-	-	-	-
SRIO_M	12	Y	12	12	12	12	Y	Y
PCIe_Master	12	Y	12	12	12	12	Y	Y
Network Coprocessor Packet DMA	-	-	-	-	-	-	-	-
MSMC_Data_Master	-	-	-	-	-	-	-	-
QM_SS Packet DMA	-	-	-	-	-	-	-	-
QM_SS Second	-	-	-	-	-	-	-	-
DebugSS_Master	12	Y	12	12	12	12	Y	Y
FFTC	-	-	-	-	-	-	-	-
RAC_BE0	-	-	-	-	-	-	-	-
RAC_BE1	-	-	-	-	-	-	-	-
AIF_Master	-	-	-	-	-	-	-	-
TAC_FE	-	-	-	-	-	-	-	-
EDMA3CC0	-	-	-	-	-	-	-	-
EDMA3CC1	-	-	-	-	-	-	-	-
EDMA3CC2	-	-	-	-	-	-	-	-
CorePac0_CFG	Y	Y	Y	Y	Y	Y	Y	Y
CorePac1_CFG	Y	Y	Y	Y	Y	Y	Y	Y
CorePac2_CFG	Y	Y	Y	Y	Y	Y	Y	Y
CorePac3_CFG	Y	Y	Y	Y	Y	Y	Y	Y

End of Table 4-3

4.3 TeraNet Switch Fabric Connections

The figures below show the connections between masters and slaves through various sections of the TeraNet.

Figure 4-1 TeraNet 3A

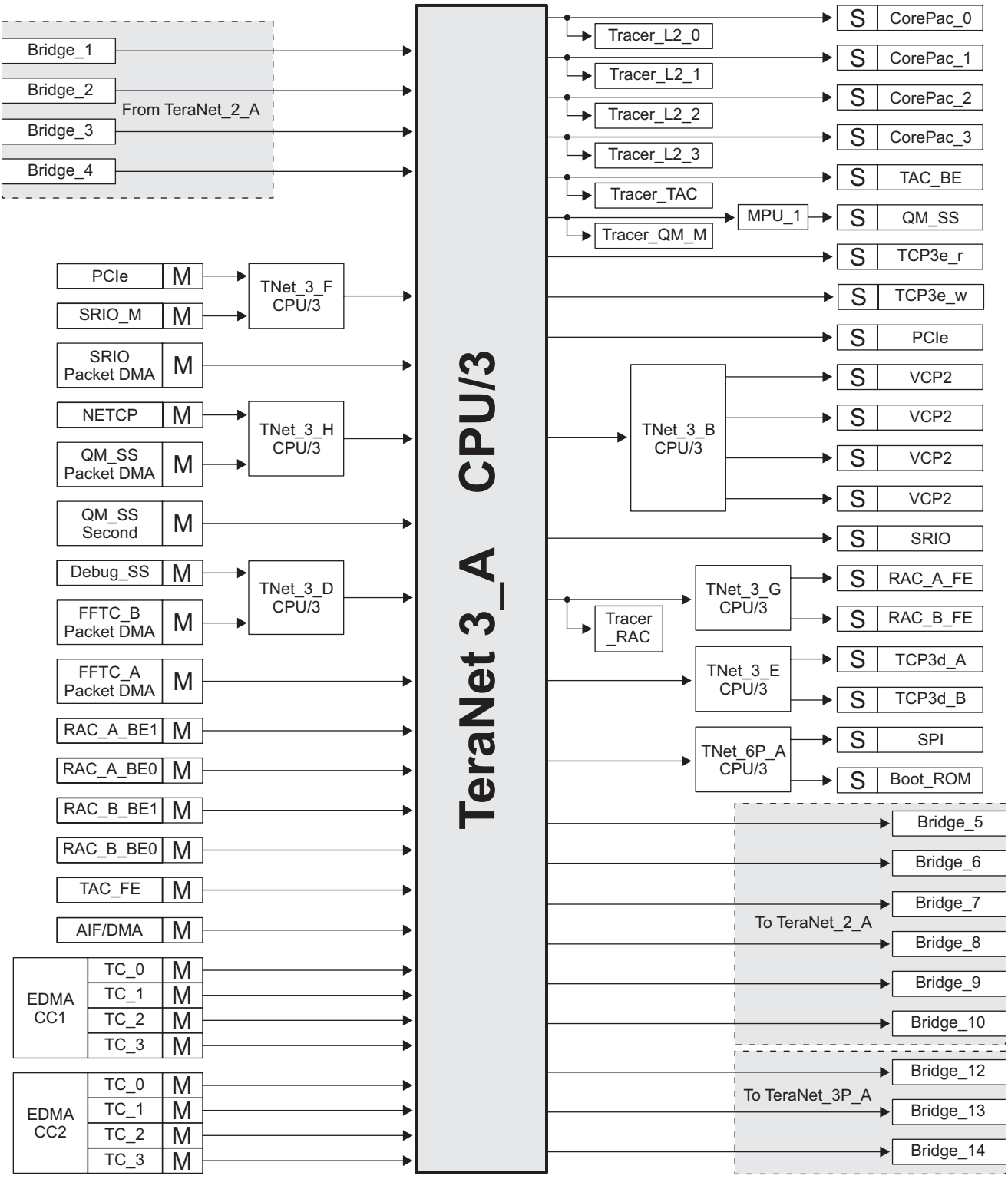


Figure 4-2 TeraNet 2A

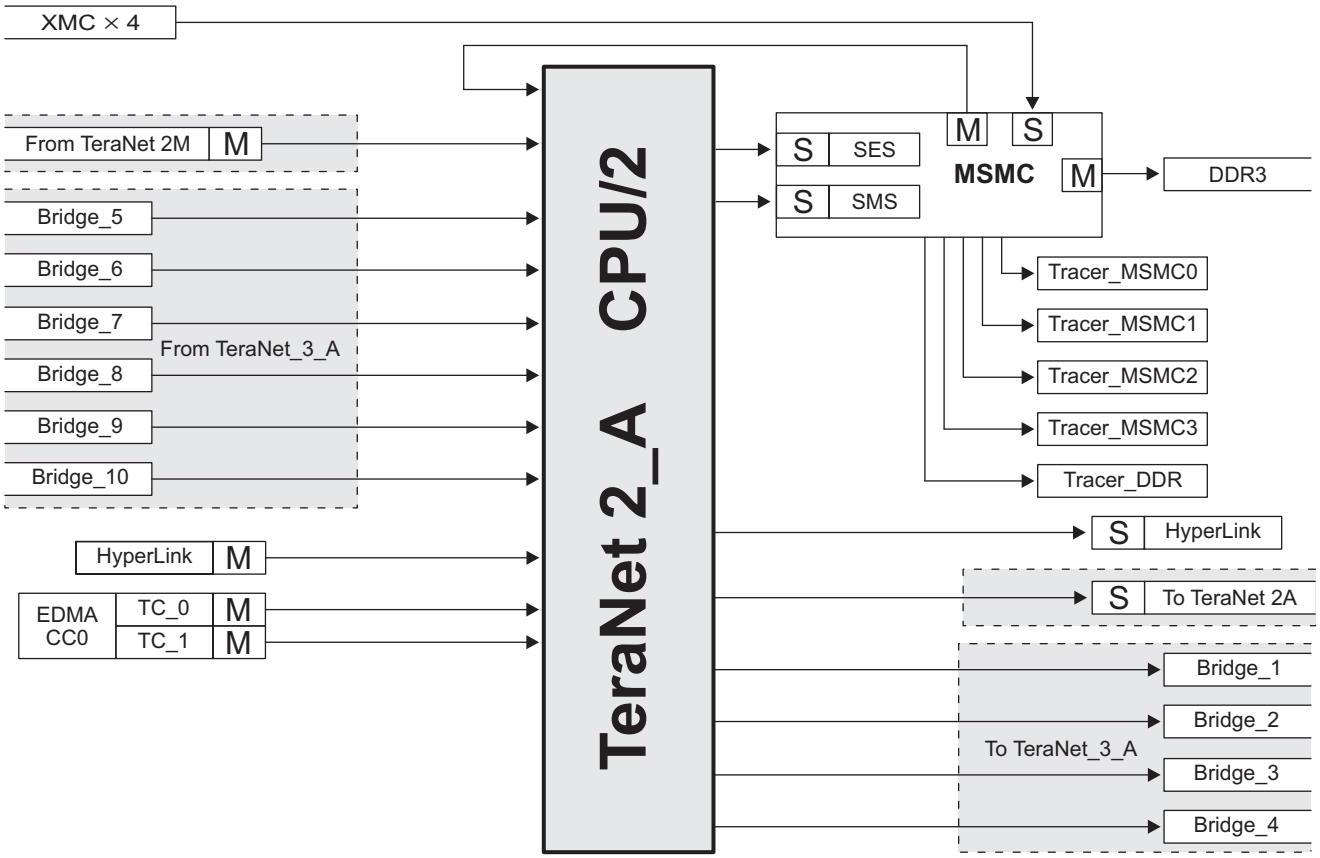


Figure 4-3 TeraNet 3P and 3M and 2M

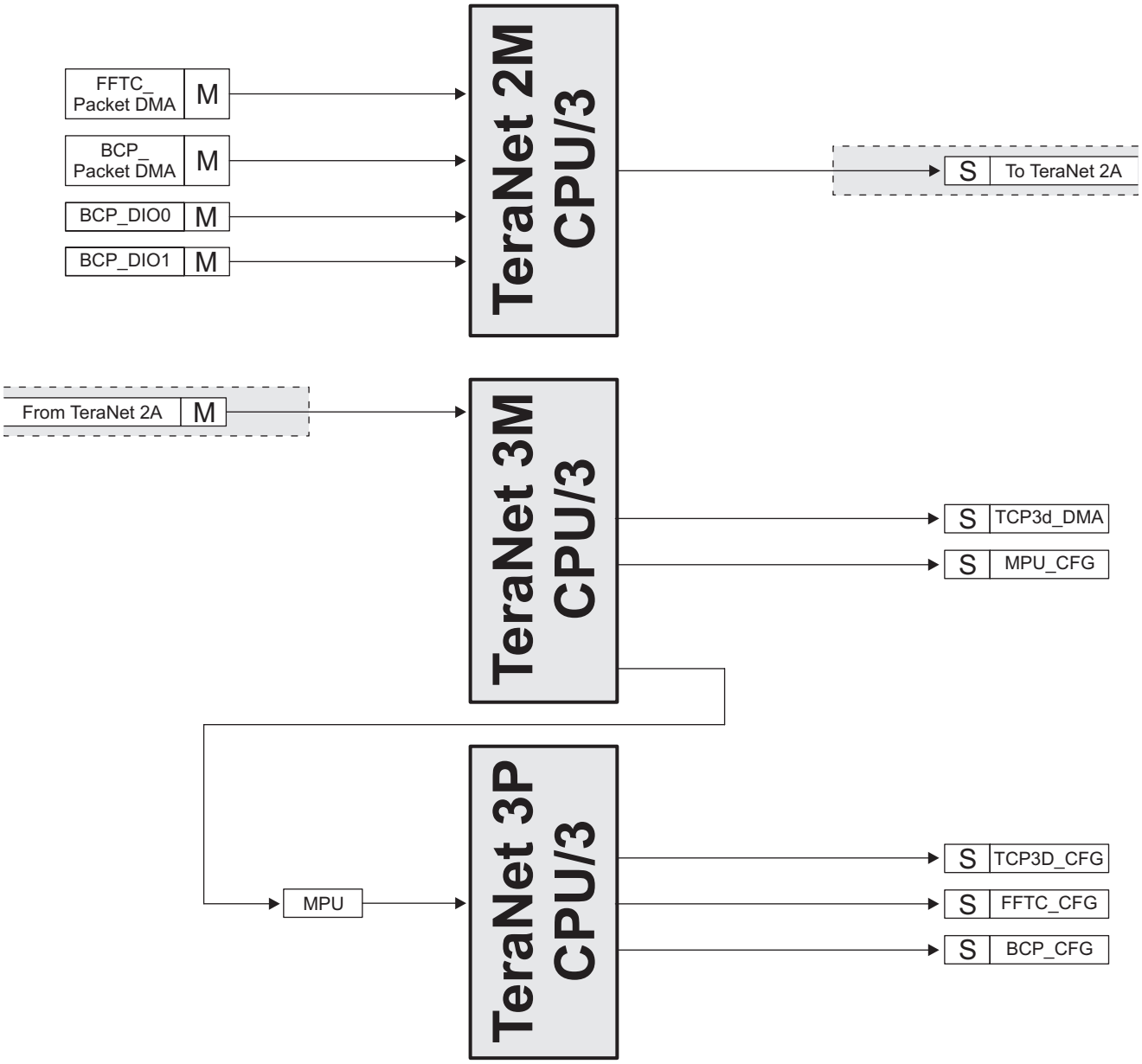
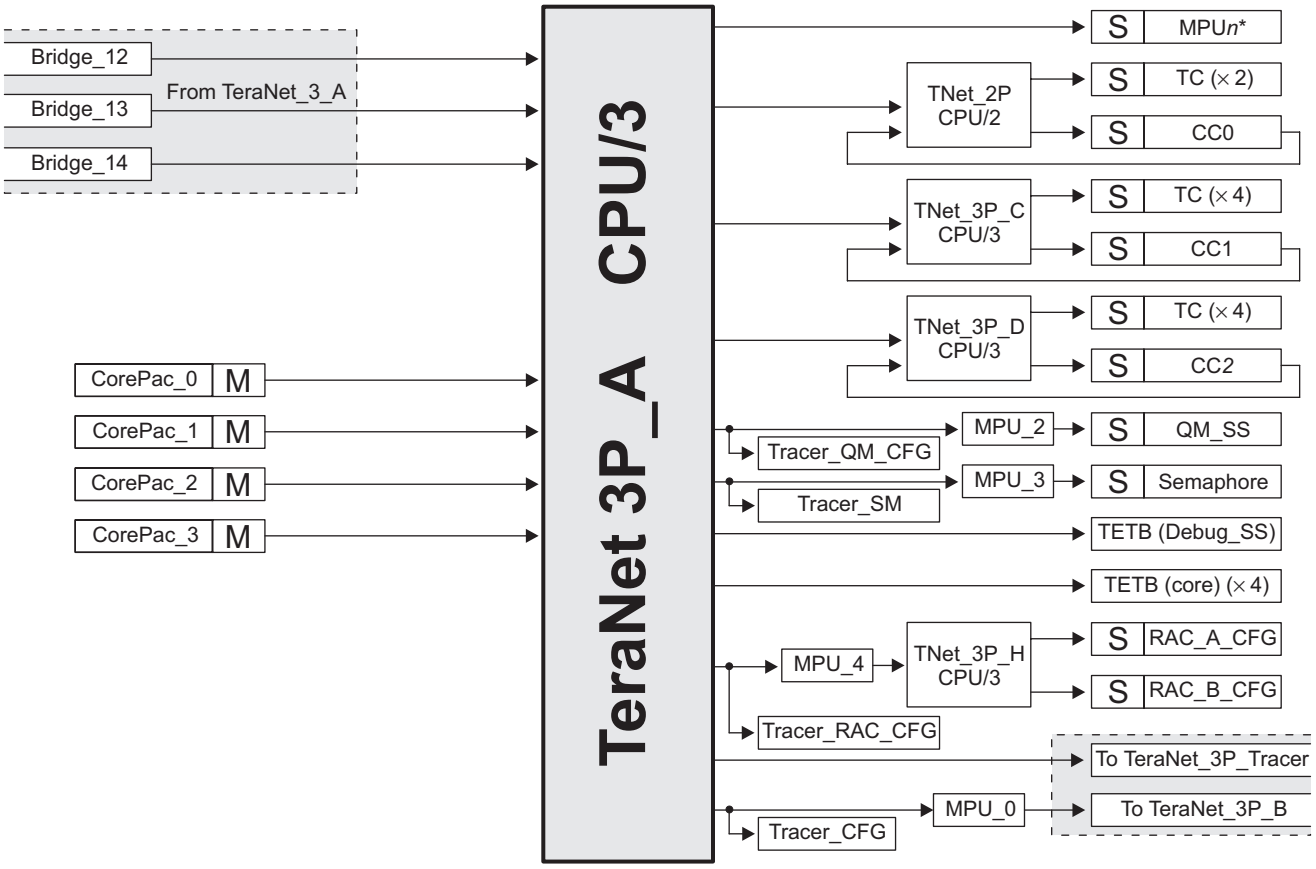


Figure 4-4 TeraNet 3P_A



* *n* indicates the number of MPUs present in the specific device.

Figure 4-5 TeraNet 3P_B

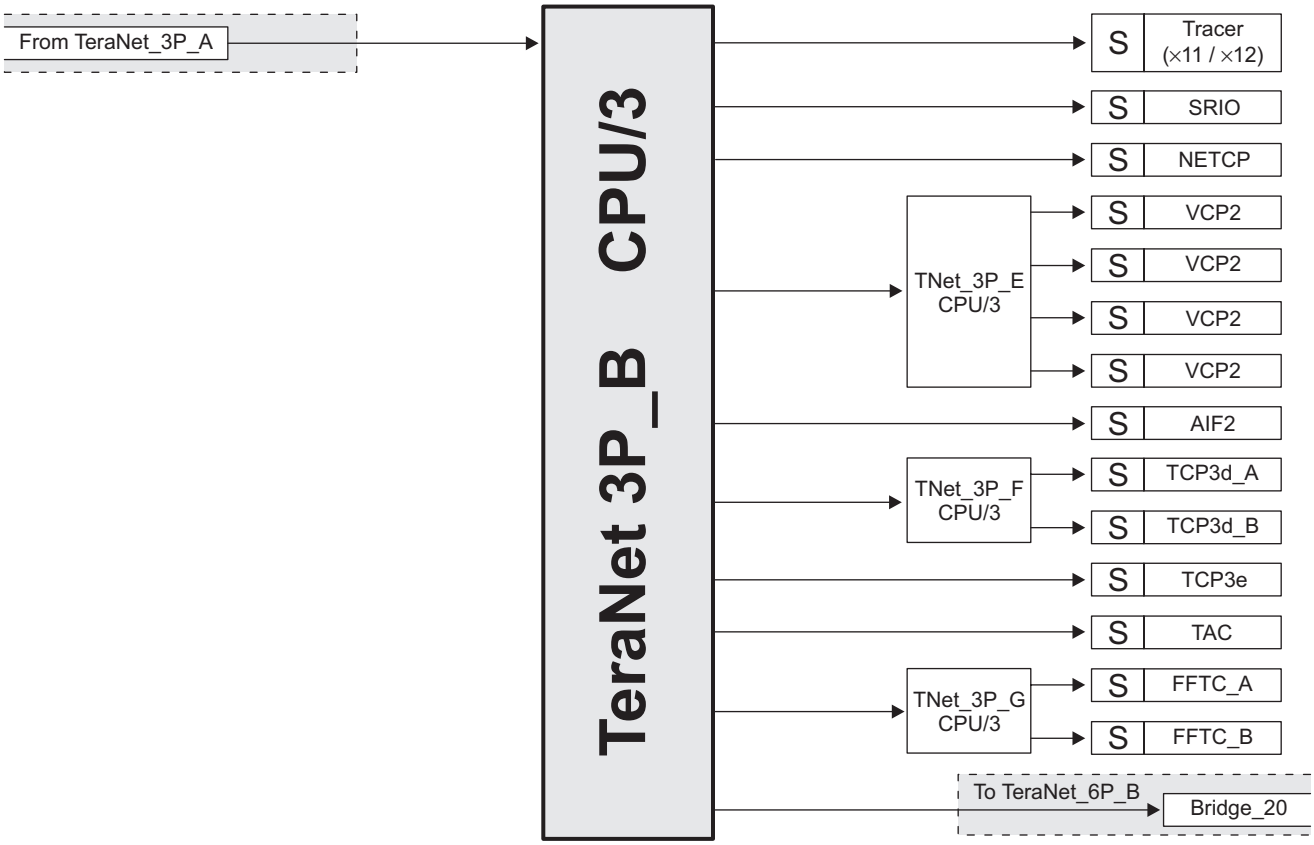
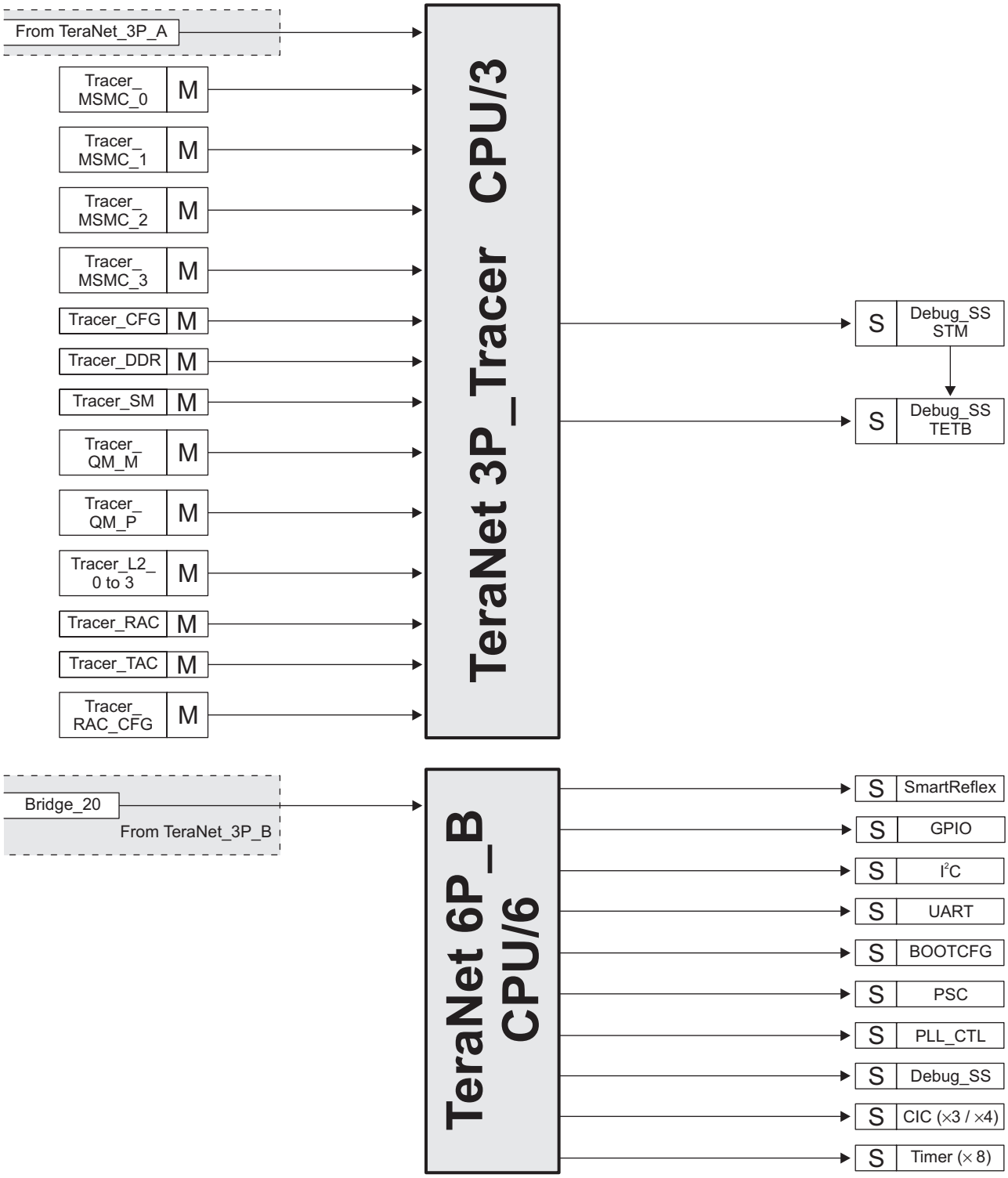


Figure 4-6 TeraNet 6P_B and 3P_Tracer



4.4 Bus Priorities

The priority level of all master peripheral traffic is defined at the TeraNet boundary. User programmable priority registers will be present to allow software configuration of the data traffic through the TeraNet. Note that a lower number means higher priority - PRI = 000b = urgent, PRI = 111b = low.

All other masters provide their priority directly and do not need a default priority setting. Examples include the CorePacs, whose priorities are set through software in the UMC control registers. All the Packet DMA based peripherals also have internal registers to define the priority level of their initiated transactions.

The Packet DMA secondary port is one master port that does not have priority allocation register inside the IP. The priority level for transaction from this master port is described by PKTDMA_PRI_ALLOC register in Figure 4-7 and Table 4-4.

Figure 4-7 Packed DMA Priority Allocation Register (PKTDMA_PRI_ALLOC)

31	3	2	0
Reserved		PKTDMA_PRI	
R/W-00000000000000000000001000011		RW-000	

Legend: R = Read only; R/W = Read/Write; -*n* = value after reset

Table 4-4 Packed DMA Priority Allocation Register Field Descriptions

Bit	Field	Description
31-3	Reserved	Reserved.
2-0	PKDTDMA_PRI	Control the priority level for the transactions from packet DMA master port, which access the external linking RAM.
End of Table 4-4		

For all other modules, see the respective User Guides in 2.9 "Related Documentation from Texas Instruments" on page 66 for programmable priority registers.

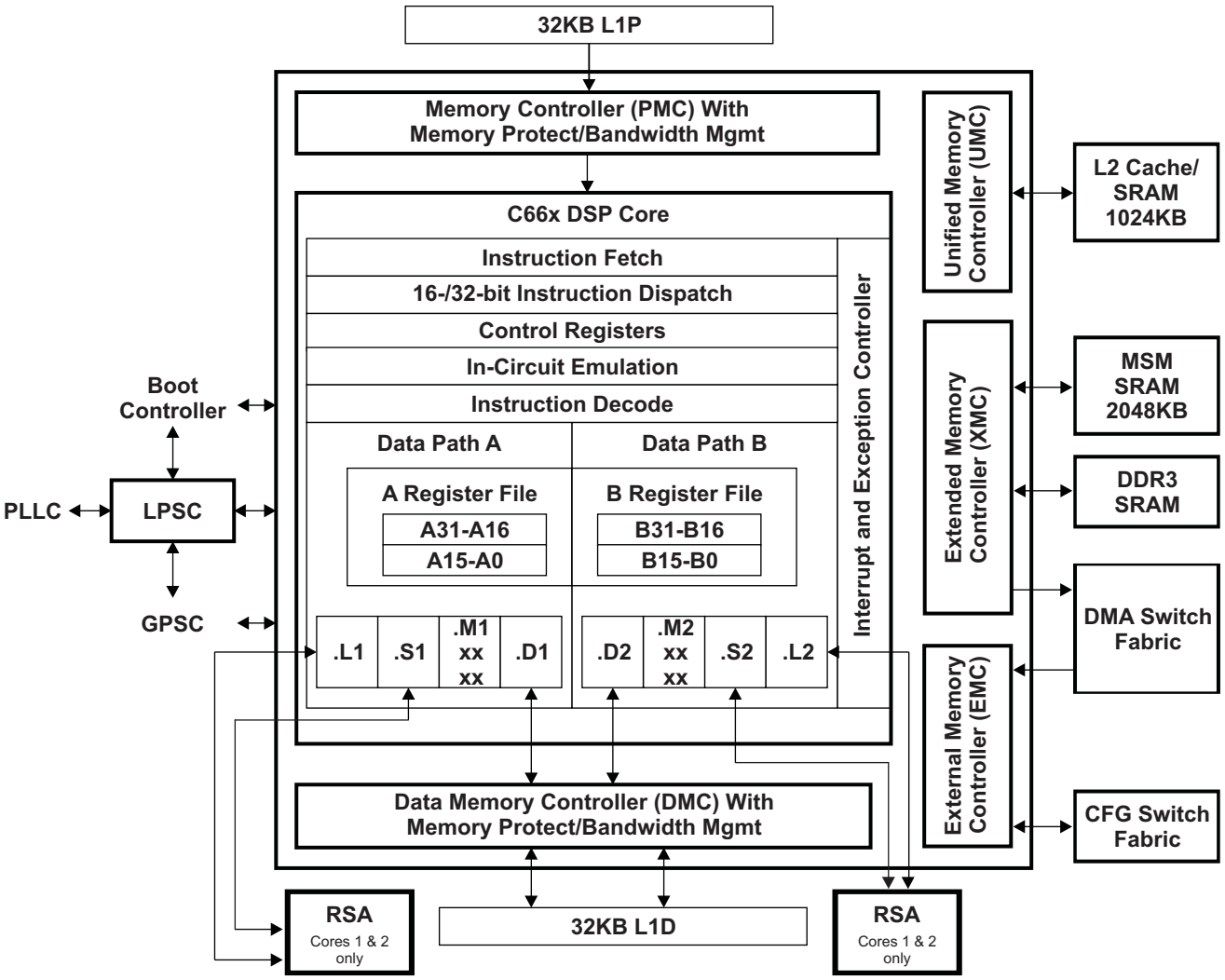
5 C66x CorePac

The C66x CorePac consists of several components:

- The C66x DSP core
- Level-one and level-two memories (L1P, L1D, L2)
- RSA accelerator (on cores 1 and 2 only)
- Data Trace Formatter (DTF)
- Embedded Trace Buffer (ETB)
- Interrupt controller
- Power-down controller
- External memory controller
- Extended memory controller
- A dedicated power/sleep controller (LPSC)

The C66x CorePac also provides support for memory protection and bandwidth management (for resources local to the CorePac). Figure 5-1 shows a block diagram of the C66x CorePac.

Figure 5-1 C66x CorePac Block Diagram



For more detailed information on the C66x CorePac in the C6670 device, see the *C66x CorePac User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

5.1 Memory Architecture

Each CorePac of the TMS320C6670 device contains a 1024KB level-2 memory (L2), a 32KB level-1 program memory (L1P), and a 32KB level-1 data memory (L1D). The device also contain a 2048KB multicore shared memory (MSM). All memory on the C6670 has a unique location in the memory map (see Table 2-2 "Memory Map Summary" on page 21.

After device reset, L1P and L1D cache are configured as all cache, by default. The L1P and L1D cache can be reconfigured via software through the L1PMODE field of the L1P Configuration Register (L1PMODE) and the L1DMODE field of the L1D Configuration Register (L1DCFG) of the C66x CorePac. L1D is a two-way set-associative cache, while L1P is a direct-mapped cache.

The on-chip bootloader changes the reset configuration for L1P and L1D. For more information, see the *Bootloader for the C66x DSP User Guide* in 2.9 "Related Documentation from Texas Instructions" on page 66.

For more information on the operation L1 and L2 caches, see the *C66x DSP Cache User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

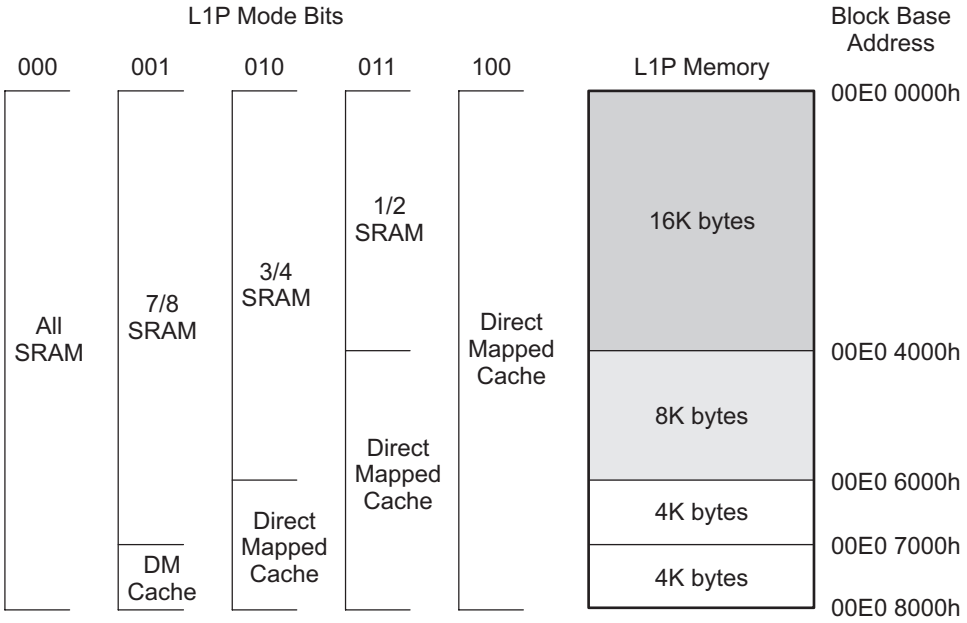
5.1.1 L1P Memory

The L1P memory configuration for the C6670 device is as follows:

- Region 0 size is 0K bytes (disabled)
- Region 1 size is 32K bytes with no wait states

Figure 5-2 shows the available SRAM/cache configurations for L1P.

Figure 5-2 L1P Memory Configurations



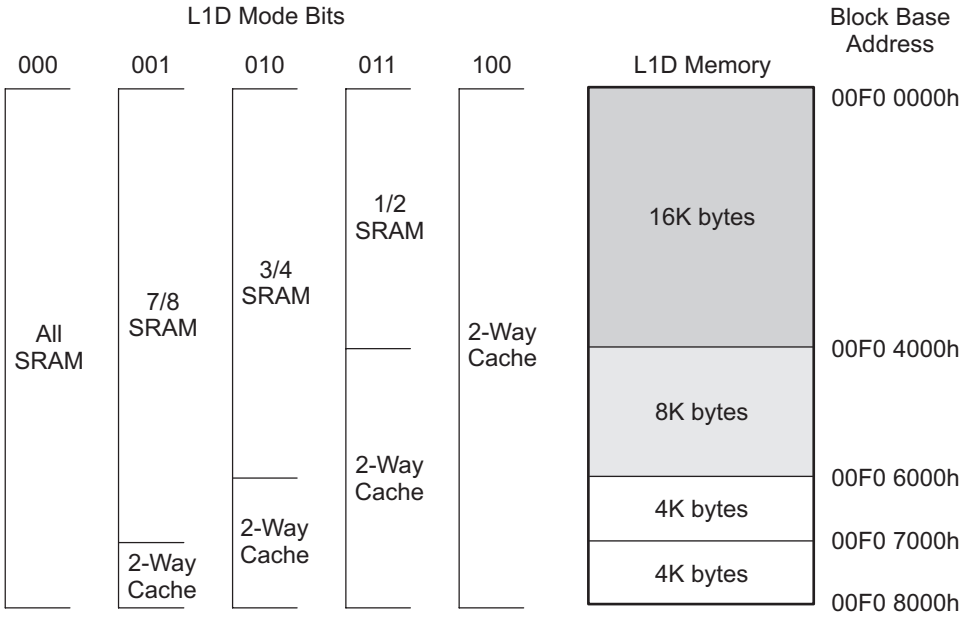
5.1.2 L1D Memory

The L1D memory configuration for the C6670 device is as follows:

- Region 0 size is 0K bytes (disabled)
- Region 1 size is 32K bytes with no wait states

Figure 5-3 shows the available SRAM/cache configurations for L1D.

Figure 5-3 L1D Memory Configurations



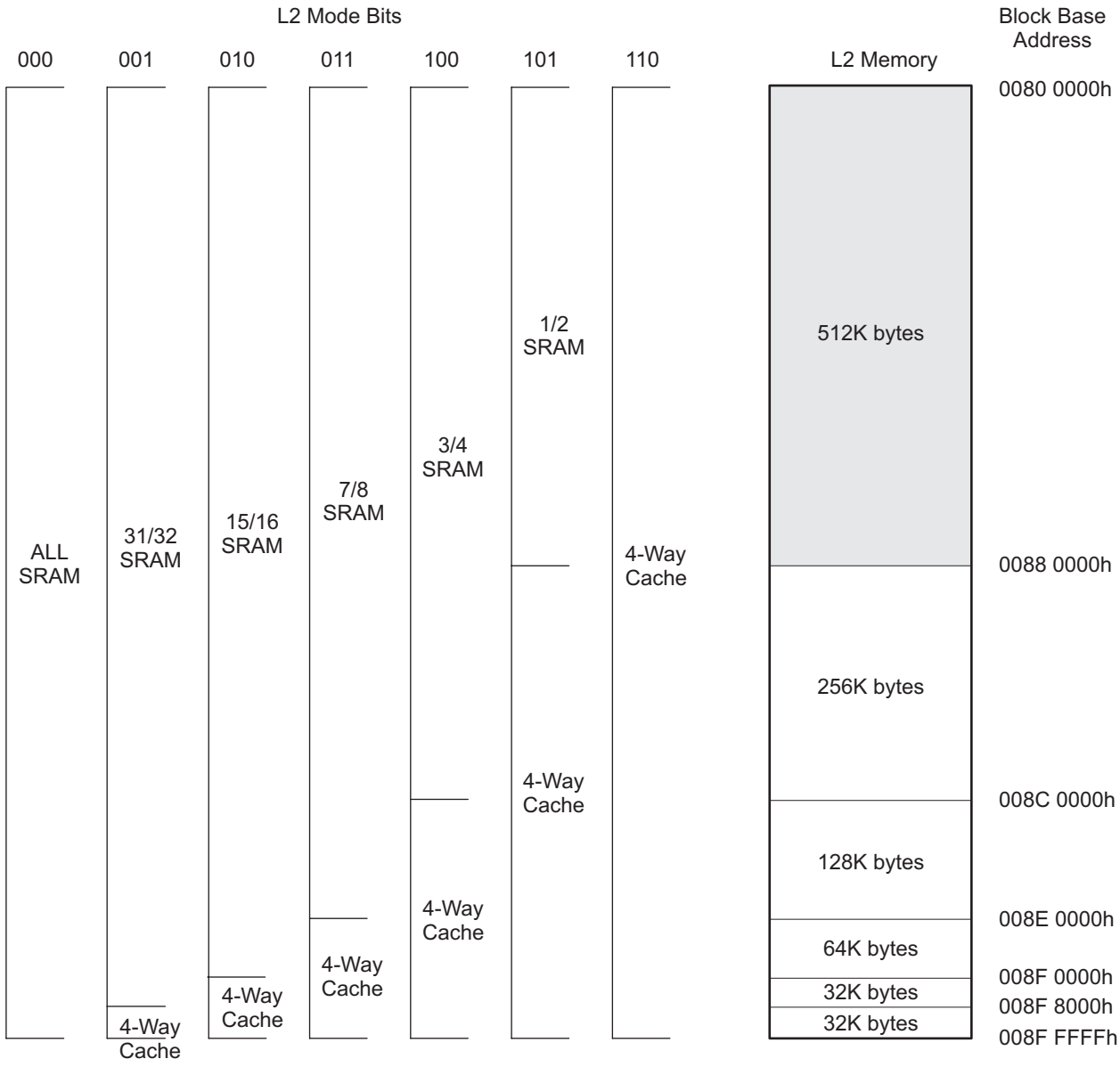
5.1.3 L2 Memory

The L2 memory configuration for the C6670 device is as follows:

- Total memory size is 4096KB
- Each CorePac contains 1024KB of memory
- Local starting address for each CorePac is 0080 0000h

L2 memory can be configured as all SRAM, all 4-way set-associative cache, or a mix of the two. The amount of L2 memory that is configured as cache is controlled through the L2MODE field of the L2 Configuration Register (L2CFG) of the C66x CorePac. Figure 5-4 shows the available SRAM/cache configurations for L2. By default, L2 is configured as all SRAM after device reset.

Figure 5-4 L2 Memory Configurations



Global addresses that are accessible to all masters in the system are in all memory local to the processors. In addition, local memory can be accessed directly by the associated processor through aliased addresses, where the eight MSBs are masked to 0. The aliasing is handled within the CorePac and allows for common code to be run unmodified on multiple cores. For example, address location 0x10800000 is the global base address for CorePac0's L2 memory. CorePac0 can access this location by either using 0x10800000 or 0x00800000. Any other master on the device must use 0x10800000 only. Conversely, 0x00800000 can by used by any of the four CorePacs as their own L2 base addresses. For CorePac0, as mentioned, this is equivalent to 0x10800000, for CorePac1 this is equivalent to 0x11800000, and for CorePac2 this is equivalent to 0x12800000. Local addresses should be used only for shared code or data, allowing a single image to be included in memory. Any code/data targeted to a specific core, or a memory region allocated during run-time by a particular CorePac should always use the global address only.

5.1.4 MSM SRAM

The MSM SRAM configuration for the C6670 device is as follows:

- Memory size is 2048KB
- The MSM can be configured as shared L2 or shared L3 memory
- Allows extension of external addresses from 2GB to up to 8GB
- Has built in memory protection features

The MSM SRAM is always configured as all SRAM. When configured as a shared L2, its contents can be cached in L1P and L1D. When configured in shared L3 mode, it's contents can be cached in L2 also. For more details on external memory address extension and memory protection features, see the *Multicore Shared Memory Controller (MSMC) for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

5.1.5 L3 Memory

The L3 ROM on the device is 128KB. The ROM contains software used to boot the device. There is no requirement to block accesses from this portion to the ROM.

5.2 Memory Protection

Memory protection allows an operating system to define who or what is authorized to access L1D, L1P, and L2 memory. To accomplish this, the L1D, L1P, and L2 memories are divided into pages. There are 16 pages of L1P (2KB each), 16 pages of L1D (2KB each), and 32 pages of L2 (32KB each). The L1D, L1P, and L2 memory controllers in the C66x CorePac are equipped with a set of registers that specify the permissions for each memory page.

Each page may be assigned with fully orthogonal user and supervisor read, write, and execute permissions. In addition, a page may be marked as either (or both) locally accessible or globally accessible. A local access is a direct DSP access to L1D, L1P, and L2, while a global access is initiated by a DMA (either IDMA or the EDMA3) or by other system masters. Note that EDMA or IDMA transfers programmed by the DSP count as global accesses. On a secure device, pages can be restricted to secure access only (default) or opened up for public, non-secure access.

The DSP and each of the system masters on the device are all assigned a privilege ID. It is only possible to specify whether memory pages are locally or globally accessible.

The AIDx and LOCAL bits of the memory protection page attribute registers specify the memory page protection scheme, see Table 5-1.

Table 5-1 Available Memory Page Protection Schemes

AIDx (1) Bit	Local Bit	Description
0	0	No access to memory page is permitted.
0	1	Only direct access by DSP is permitted.
1	0	Only accesses by system masters and IDMA are permitted (includes EDMA and IDMA accesses initiated by the DSP).
1	1	All accesses permitted.
End of Table 5-1		

1 x = 0, 1, 2, 3, 4, 5

Faults are handled by software in an interrupt (or an exception, programmable within the CorePac interrupt controller) service routine. A DSP or DMA access to a page without the proper permissions will:

- Block the access — reads return 0, writes are ignored
- Capture the initiator in a status register — ID, address, and access type are stored
- Signal event to DSP interrupt controller

The software is responsible for taking corrective action to respond to the event and resetting the error status in the memory controller. For more information on memory protection for L1D, L1P, and L2, see the *C66x CorePac User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

5.3 Bandwidth Management

When multiple requestors contend for a single C66x CorePac resource, the conflict is resolved by granting access to the highest priority requestor. The following four resources are managed by the Bandwidth Management control hardware:

- Level 1 Program (L1P) SRAM/Cache
- Level 1 Data (L1D) SRAM/Cache
- Level 2 (L2) SRAM/Cache
- Memory-mapped registers configuration bus

The priority level for operations initiated within the C66x CorePac are declared through registers in the CorePac. These operations are:

- DSP-initiated transfers
- User-programmed cache coherency operations
- IDMA-initiated transfers

The priority level for operations initiated outside the CorePac by system peripherals is declared through the Priority Allocation Register (PRI_ALLOC), see Section 4.4 "Bus Priorities" on page 97. System peripherals with no fields in PRI_ALLOC have their own registers to program their priorities.

More information on the bandwidth management features of the CorePac can be found in the *C66x CorePac Reference Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

5.4 Power-Down Control

The C66x CorePac supports the ability to power-down various parts of the CorePac. The power-down controller (PDC) of the CorePac can be used to power down L1P, the cache control hardware, the DSP, and the entire CorePac. These power-down features can be used to design systems for lower overall system power requirements.

Note—The C6670 does not support power-down modes for the L2 memory at this time.

More information on the power-down features of the C66x CorePac can be found in the *C66x CorePac Reference Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66

5.5 CorePac Revision

The version and revision of the C66x CorePac can be read from the CorePac Revision ID Register (MM_REVID) located at address 0181 2000h. The MM_REVID register is shown in Table 5-2 and described in Table 5-2. The C66x CorePac revision is dependant on the silicon revision being used.

Figure 5-5 CorePac Revision ID Register (MM_REVID)

31	16	15	0
VERSION		REVISION	
R-n		R-n	

Legend: R = Read only; R/W = Read/Write; *-n* = value after reset

Table 5-2 CorePac Revision ID Register (MM_REVID) Field Descriptions

Bit	Name	Value	Description
31-16	VERSION	xxxxh	Version of the C66x CorePac implemented on the device will depend on the silicon being used.
15-0	REVISION	0000h	Revision of the C66x CorePac version implemented on this device.
End of Table 5-2			

5.6 C66x CorePac Register Descriptions

See the *C66x CorePac User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66 for register offsets and definitions.

6 Device Operating Conditions

6.1 Absolute Maximum Ratings

Table 6-1 Absolute Maximum Ratings [1]
Over Operating Case Temperature Range (Unless Otherwise Noted)

Supply voltage range [2]:	CVDD	-0.3 V to 1.3 V
	CVDD1	-0.3 V to 1.3 V
	DVDD15	-0.3 V to 2.45 V
	DVDD18	-0.3 V to 2.45 V
	VREFSSTL	0.49 × DVDD15 to 0.51 × DVDD15
	VDDT1, VDDT2, VDDT3	-0.3 V to 1.3 V
	VDDR1, VDDR2, VDDR3 VDDR4, VDDR5, VDDR6	-0.3 V to 2.45 V
	AVDDA1, AVDDA2, AVDDA3	-0.3 V to 2.45 V
	VSS Ground	0 V
Input voltage (V_I) range:	LVCMOS (1.8 V)	-0.3 V to DVDD18+0.3 V
	DDR3	-0.3 V to 2.45 V
	I^2C	-0.3 V to 2.45 V
	LVDS	-0.3 V to DVDD18+0.3 V
	LJCB	-0.3 V to 1.3 V
	SerDes	-0.3 V to CVDD1+0.3 V
Output voltage (V_O) range:	LVCMOS (1.8 V)	-0.3 V to DVDD18+0.3 V
	DDR3	-0.3 V to 2.45 V
	I^2C	-0.3 V to 2.45 V
	SerDes	-0.3 V to CVDD1+0.3 V
Operating case temperature range, T_C:	Commercial	0°C to 100°C
	Extended	-40°C to 100°C
ESD stress voltage, V_{ESD} [3]	HBM (human body model) [4]	±1000 V
	CDM (charged device model) [5]	±250 V
Overshoot/undershoot [6]	LVCMOS (1.8 V)	20% overshoot/undershoot for 20% of signal duty cycle
	DDR3	
	I^2C	
Storage temperature range, T_{stg}:		-65°C to 150°C
End of Table 6-1		

1 Stresses beyond those listed under *absolute maximum ratings* may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under *recommended operating conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

2 All voltage values are with respect to V_{SS}.

3 Electrostatic discharge (ESD) to measure device sensitivity/immunity to damage caused by electrostatic discharges into the device.

4 Level listed above is the passing level per ANSI/ESDA/JEDEC JS-001-2010. JEDEC document JEP155 states that 500 V HBM allows safe manufacturing with a standard ESD control process, and manufacturing with less than 500 V HBM is possible if necessary precautions are taken. Pins listed as 1000 V may actually have higher performance.

5 Level listed above is the passing level per EIA-JEDEC JESD22-C101E. JEDEC document JEP157 states that 250 V CDM allows safe manufacturing with a standard ESD control process. Pins listed as 250 V may actually have higher performance.

6 Overshoot/Undershoot percentage relative to I/O operating values - for example the maximum overshoot value for 1.8V LVCMOS signals is DVDD18 + 0.20 × DVDD18 and maximum undershoot value would be V_{SS} - 0.20 × DVDD18

6.2 Recommended Operating Conditions

Table 6-2 Recommended Operating Conditions (1) (2)

			Min	Nom	Max	Unit
CVDD	SR core supply		SRVnom*0.95 (3)	0.9-1.1	SRVnom*1.05	V
CVDD1	Core supply		0.95	1	1.05	V
DVDD18	1.8-V supply I/O voltage		1.71	1.8	1.89	V
DVDD15	1.5-V supply I/O voltage		1.425	1.5	1.575	V
VREFSSTL	DDR3 reference voltage		0.49 × DVDD15	0.5 × DVDD15	0.51 × DVDD15	V
V_{DDRx} (4)	SerDes regulator supply		1.425	1.5	1.575	V
V_{DDAx}	PLL analog supply		1.71	1.8	1.89	V
V_{DDTx}	SerDes termination supply		0.95	1	1.05	V
V_{SS}	Ground		0	0	0	V
V_{IH}	High-level input voltage	LVCMOS (1.8 V)	0.65 × DVDD18			V
		I^2C	0.7 × DVDD18			V
		DDR3 EMIF	VREFSSTL + 0.1			V
V_{IL}	Low-level input voltage	LVCMOS (1.8 V)			0.35 × DVDD18	V
		DDR3 EMIF	-0.3		VREFSSTL - 0.1	V
		I^2C			0.3 × DVDD18	V
T_C	Operating case temperature	Commercial	0		100	°C
		Extended	-40		100	°C

End of Table 6-2

1 All differential clock inputs comply with the LVDS Electrical Specification, IEEE 1596.3-1996 and all SerDes I/Os comply with the XAUI Electrical Specification, IEEE 802.3ae-2002.

2 All SerDes I/Os comply with the XAUI Electrical Specification, IEEE 802.3ae-2002.

3 SRVnom refers to the unique SmartReflex core supply voltage between 0.9 V and 1.1 V set from the factory for each individual device.

4 Where x = 1, 2, 3, 4... to indicate all supplies of the same kind.

6.3 Electrical Characteristics

Table 6-3 Electrical Characteristics
Over Recommended Ranges of Supply Voltage and Operating Case Temperature (Unless Otherwise Noted)

Parameter			Test Conditions [1]	Min	Typ	Max	Unit
V_{OH}	High-level output voltage	LVCMOS (1.8 V)	$I_O = I_{OH}$	DVDD18 - 0.45			V
		DDR3		DVDD15 - 0.4			
		I^2C [2]					
V_{OL}	Low-level output voltage	LVCMOS (1.8 V)	$I_O = I_{OL}$			0.45	V
		DDR3				0.4	
		I^2C	I_O = 3 mA, pulled up to 1.8 V			0.4	
I_I [3]	Input current [DC]	LVCMOS (1.8 V)	No IPD/IPU	-5		5	µA
			Internal pullup	50	100	170	
			Internal pulldown	-170	-100	-50	
		I^2C	0.1 × DVDD18 V < V_I < 0.9 × DVDD18 V	-10		10	µA
I_{OH}	High-level output current [DC]	LVCMOS (1.8 V)				-6	mA
		DDR3				-8	
		I^2C [4]					
I_{OL}	Low-level output current [DC]	LVCMOS (1.8 V)				6	mA
		DDR3				8	
		I^2C				3	
I_{OZ} [5]	Off-state output current [DC]	LVCMOS (1.8 V)		-2		2	µA
		DDR3		-2		2	
		I^2C		-2		2	

End of Table 6-3

1 For test conditions shown as MIN, MAX, or TYP, use the appropriate value specified in the recommended operating conditions table.
2 I^2C uses open collector IOs and does not have a V_{OH} Minimum.
3 I_I applies to input-only pins and bidirectional pins. For input-only pins, I_I indicates the input leakage current. For bidirectional pins, I_I includes input leakage current and off-state (Hi-Z) output leakage current.
4 I^2C uses open collector IOs and does not have a I_{OH} Maximum.
5 I_{OZ} applies to output-only pins, indicating off-state (Hi-Z) output leakage current.

6.4 Power Supply to Peripheral I/O Mapping

Table 6-4 Power Supply to Peripheral I/O Mapping (1) (2)
Over Recommended Ranges of Supply Voltage and Operating Case Temperature (Unless Otherwise Noted)

Power Supply		I/O Buffer Type	Associated Peripheral
CVDD	Supply core voltage	LJCB	SYSCLK(P|N) PLL input buffer
			ALTCORECLK(P|N) PLL input buffer
			SRIOSGMIICLK(P|N) SerDes PLL input buffer
			DDRCLK(P|N) PLL input buffer
			PCIECLK(P|N) SerDes PLL input buffer
			MCMCLK(P|N) SerDes PLL input buffer
			PASSCLK(P|N) PLL input buffer
DVDD15	1.5-V supply I/O voltage	DDR3 (1.5 V)	All DDR3 memory controller peripheral I/O buffer
DVDD18	1.8-V supply I/O voltage	LVCMOS (1.8V)	All GPIO peripheral I/O buffer
			All JTAG and EMU peripheral I/O buffer
			All TIMER0-8 peripheral I/O buffer
			All SPI peripheral I/O buffer
			All AIF peripheral I/O buffer
			All RESETs, NMI, control peripheral I/O buffer
			All SmartReflex peripheral I/O buffer
			All Hyperlink sideband peripheral I/O buffer
			All MDIO peripheral I/O buffer
			All UART peripheral I/O buffer
		Open-drain (1.8 V)	All I^2C peripheral I/O buffer
VDDT1	Hyperlink SerDes termination and analogue front-end supply	SerDes/CML	Hyperlink SerDes CML IO buffer
VDDT2	SRIO/SGMII/PCIE SerDes termination and analog front-end supply	SerDes/CML	SRIO/SGMII/PCIE SerDes CML IO buffer
VDDT3	AIF termination and analog front-end supply	SerDes/CML	AIF SerDes CML IO buffer
End of Table 6-4			

1 Please note that this table does not attempt to describe all functions of all power supply terminals but only those whose purpose it is to power peripheral I/O buffers and clock input buffers.

2 Please see the *Hardware Design Guide for KeyStone Devices* in 2.9 "Related Documentation from Texas Instruments" on page 66 for more information about individual peripheral I/O.

7 TMS320C6670 Peripheral Information and Electrical Specifications

This chapter covers the various peripherals on the TMS320C6670 device. Peripheral-specific information, timing diagrams, electrical specifications, and register memory maps are described in this chapter.

7.1 Recommended Clock and Control Signal Transition Behavior

All clocks and control signals ***must*** transition between V_{IH} and V_{IL} (or between V_{IL} and V_{IH}) in a monotonic manner.

7.2 Power Supplies

The following sections describe the proper power-supply sequencing and timing needed to properly power on the C6670. The various power supply rails and their primary function is listed in Table 7-1.

Table 7-1 Power Supply Rails on the TMS320C6670

Name	Primary Function	Voltage	Notes
CVDD	SmartReflex core supply voltage	0.9 - 1.1 V	Variable core supply
CVDD1	Core supply voltage for memory array	1.0 V	Fixed supply at 1.0 V
VDDT1	HyperLink SerDes termination supply	1.0 V	Filtered version of CVDD1. Special considerations for noise. Filter is not needed if HyperLink is not in use.
VDDT2	SGMII/SRIO/PCIE SerDes termination supply	1.0 V	Filtered version of CVDD1. Special considerations for noise. Filter is not needed if SGMII/SRIO/PCIE is not in use.
VDDT3	AIF SerDes termination supply	1.0 V	Filtered version of CVDD1. Special considerations for noise. Filter is not needed if AIF is not in use.
DVDD15	1.5-V DDR3 IO supply	1.5 V	Fixed supply at 1.5 V
VDDR1	HyperLink SerDes regulator supply	1.5 V	Filtered version of DVDD15. Special considerations for noise. Filter is not needed if HyperLink is not in use.
VDDR2	PCIE SerDes regulator supply	1.5 V	Filtered version of DVDD15. Special considerations for noise. Filter is not needed if PCIE is not in use.
VDDR3	SGMII SerDes regulator supply	1.5 V	Filtered version of DVDD15. Special considerations for noise. Filter is not needed if SGMII is not in use.
VDDR4	SRIO SerDes regulator supply	1.5 V	Filtered version of DVDD15. Special considerations for noise. Filter is not needed if SRIO is not in use.
VDDR5	AIF SerDes regulator supply	1.5 V	Filtered version of DVDD15. Special considerations for noise. Filter is not needed if AIF is not in use.
VDDR6			
DVDD18	1.8-V IO supply	1.8 V	Fixed supply at 1.8 V
AVDDA1	Main PLL supply	1.8 V	Filtered version of DVDD18. Special considerations for noise.
AVDDA2	DDR3 PLL supply	1.8 V	Filtered version of DVDD18. Special considerations for noise.
AVDDA3	PASS PLL supply	1.8 V	Filtered version of DVDD18. Special considerations for noise.
VREFSSTL	0.75-V DDR3 reference voltage	0.75 V	Should track the 1.5-V supply. Use 1.5 V as source.
VSS	Ground	GND	Ground
End of Table 7-1			

7.2.1 Power-Up Sequencing

This section defines the requirements for a power up sequencing from a power-on reset condition. There are two acceptable power sequences for the device. The first sequence stipulates the core voltages starting before the IO voltages as shown below.

1. CVDD
2. CVDD1, VDDT1-3
3. DVDD18, AVDD1, AVDD2
4. DVDD15, VDDR1-6

The second sequence provides compatibility with other TI processors with the IO voltage starting before the core voltages as shown below.

1. DVDD18, AVDD1, AVDD2
2. CVDD
3. CVDD1, VDDT1-3
4. DVDD15, VDDR1-6

The clock input buffers for SYSCLK, ALTCORECLK, DDRCLK, PASSCLK, SRIOSGMIICLK, PCIECLK, and MCMCLK use CVDD as a supply voltage. These clock inputs are not failsafe and must be held in a high-impedance state until CVDD is at a valid voltage level. Driving these clock inputs high before CVDD is valid could cause damage to the device. Once CVDD is valid, it is acceptable that the P and N legs of these clocks may be held in a static state (either high and low or low and high) until a valid clock frequency is needed at that input. To avoid internal oscillation, the clock inputs should be removed from the high impedance state shortly after CVDD is present.

If a clock input is not used, it must be held in a static state. To accomplish this, the N leg should be pulled to ground through a 1-kΩ resistor. The P leg should be tied to CVDD to ensure it will not have any voltage present until CVDD is active. Connections to the IO cells powered by DVDD18 and DVDD15 are not failsafe and should not be driven high before these voltages are active. Driving these IO cells high before DVDD18 or DVDD15 are valid could cause damage to the device.

The device initialization is divided into two phases. The first phase consists of the time period from the activation of the first power supply until the point at which all supplies are active and at a valid voltage level. Either of the sequencing scenarios described above can be implemented during this phase. The figures below show both the core-before-IO voltage sequence and the IO-before-core voltage sequence. $\overline{\text{POR}}$ must be held low for the entire power stabilization phase.

This is followed by the device initialization phase. The rising edge of $\overline{\text{POR}}$ followed by the rising edge of $\overline{\text{RESETFULL}}$ will trigger the end of the initialization phase but both must be inactive for the initialization to complete. $\overline{\text{POR}}$ must always go inactive before $\overline{\text{RESETFULL}}$ goes inactive as described below. SYSCLK1 in the following section refers to the clock that is used by the CorePac, see Figure 7-7 for more details.

7.2.1.1 Core-Before-IO Power Sequencing

Figure 7-1 shows the power sequencing and reset control of the TMS320C6670 for device initialization. $\overline{\text{POR}}$ may be removed after the power has been stable for the required 100 µsec. $\overline{\text{RESETFULL}}$ must be held low for a period after the rising edge of $\overline{\text{POR}}$, but may be held low for longer periods if necessary. The configuration bits shared with the GPIO pins will be latched on the rising edge of $\overline{\text{RESETFULL}}$ and must meet the setup and hold times specified. SYSCLK1 must always be active before $\overline{\text{POR}}$ can be removed. Core-before-IO power sequencing is defined in Table 7-2.

Note—TI recommends a maximum of 100 ms between one power rail being valid, and the next power rail in the sequence starting to ramp.

Figure 7-1 Core Before IO Power Sequencing

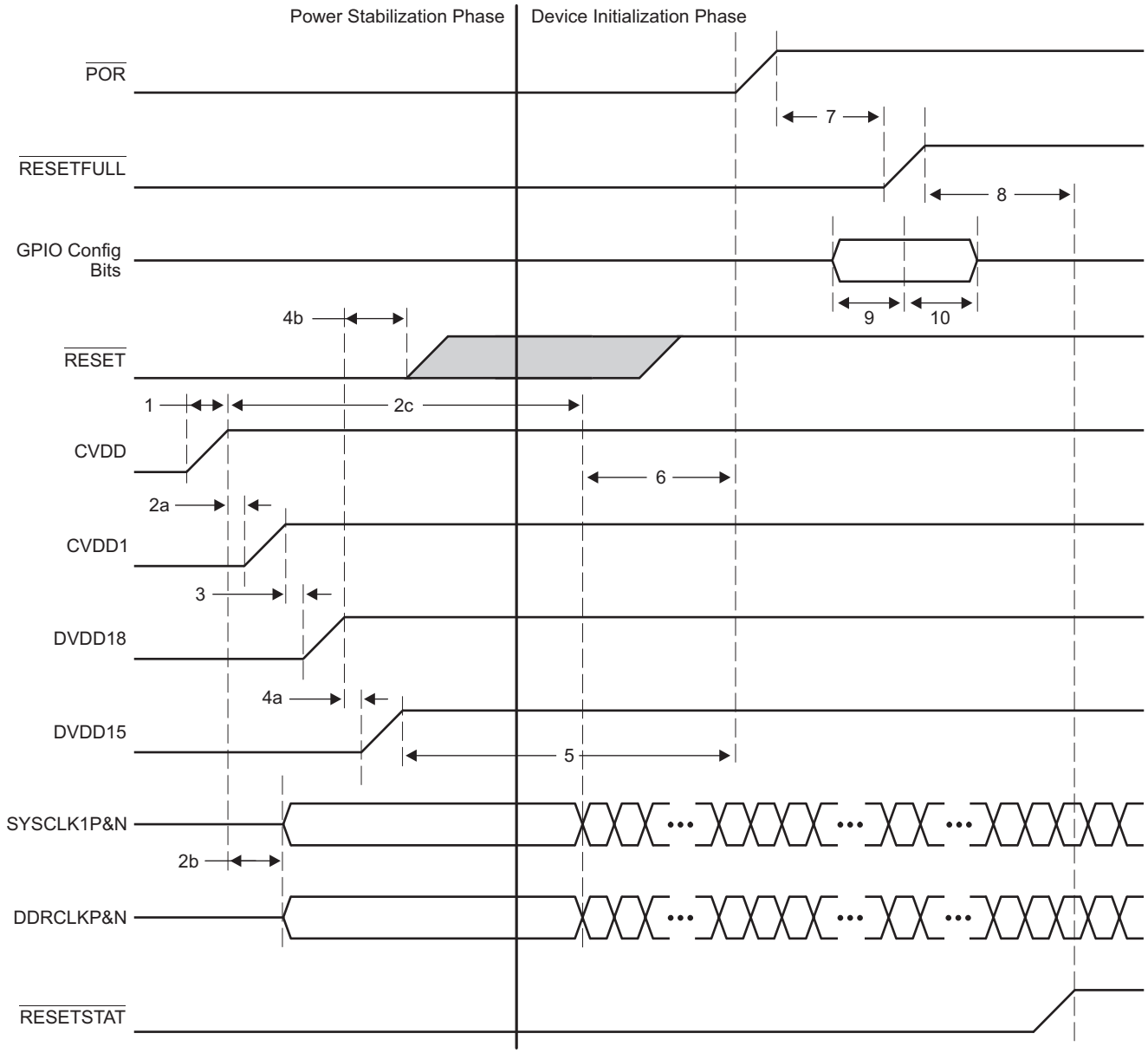


Table 7-2 Core Before IO Power Sequencing

Time	System State
1	**Begin Power Stabilization Phase** • CVDD (core AVS) ramps up. • $\overline{\text{POR}}$ must be held low through the power stabilization phase. Because $\overline{\text{POR}}$ is low, all the core logic that has async reset (created from $\overline{\text{POR}}$) is put into the reset state.
2a	• CVDD1 (core constant) ramps at the same time or shortly following CVDD. Although ramping CVDD1 and CVDD simultaneously is permitted, the voltage for CVDD1 ***must never exceed*** CVDD until after CVDD has reached a valid voltage. • The purpose of ramping up the core supplies close to each other is to reduce crowbar current. CVDD1 should trail CVDD as this will ensure that the WLs in the memories are turned off and there is no current through the memory bit cells. If, however, CVDD1 (core constant) ramps up before CVDD (core AVS), then the worst-case current could be on the order of twice the specified draw of CVDD1.
2b	• Once CVDD is valid, the clock drivers should be enabled. Although the clock inputs are not necessary at this time, they should either be driven with a valid clock or be held in a static state with one leg high and one leg low.
2c	• The DDRCLK and SYSCLK1 may begin to toggle anytime between when CVDD is at a valid level and the setup time before $\overline{\text{POR}}$ goes high specified by t6.
3	• Filtered versions of 1.8 V can ramp simultaneously with DVDD18. • $\overline{\text{RESETSTAT}}$ is driven low once the DVDD18 supply is available. • All LVCMOS input and bidirectional pins must not be driven or pulled high until DVDD18 is present. Driving an input or bidirectional pin before DVDD18 is valid could cause damage to the device.
4a	• DVDD15 (1.5 V) supply is ramped up following DVDD18. Although ramping DVDD18 and DVDD15 simultaneously is permitted, the voltage for DVDD15 ***must never exceed*** DVDD18.
4b	• $\overline{\text{RESET}}$ may be driven high any time after DVDD18 is at a valid level. In a $\overline{\text{POR}}$-controlled boot, $\overline{\text{RESET}}$ must be high before $\overline{\text{POR}}$ is driven high.
5	• $\overline{\text{POR}}$ must continue to remain low for at least 100 µs after power has stabilized. End power stabilization phase
6	• Device initialization requires 500 SYSCLK1 periods after the Power Stabilization Phase. The maximum clock period is 33.33 nsec, so a delay of an additional 16 µs is required before a rising edge of $\overline{\text{POR}}$. The clock must be active during the entire 16 µs.
7	• $\overline{\text{RESETFULL}}$ must be held low for at least 24 transitions of the SYSCLK1 after $\overline{\text{POR}}$ has stabilized at a high level.
8	• The rising edge of the $\overline{\text{RESETFULL}}$ will remove the reset to the efuse farm allowing the scan to begin. • Once device initialization and the efuse farm scan are complete, the $\overline{\text{RESETSTAT}}$ signal is driven high. This delay will be 10000 to 50000 clock cycles. End device initialization phase
9	• GPIO configuration bits must be valid for at least 12 transitions of the SYSCLK1 before the rising edge of $\overline{\text{RESETFULL}}$
10	• GPIO configuration bits must be held valid for at least 12 transitions of the SYSCLK1 after the rising edge of $\overline{\text{RESETFULL}}$
End of Table 7-2	

7.2.1.2 IO-Before-Core Power Sequencing

The timing diagram for IO-before-core power sequencing is shown in Figure 7-2 and defined in Table 7-3.

Note—TI recommends a maximum of 100 ms between one power rail being valid, and the next power rail in the sequence starting to ramp.

Figure 7-2 IO Before Core Power Sequencing

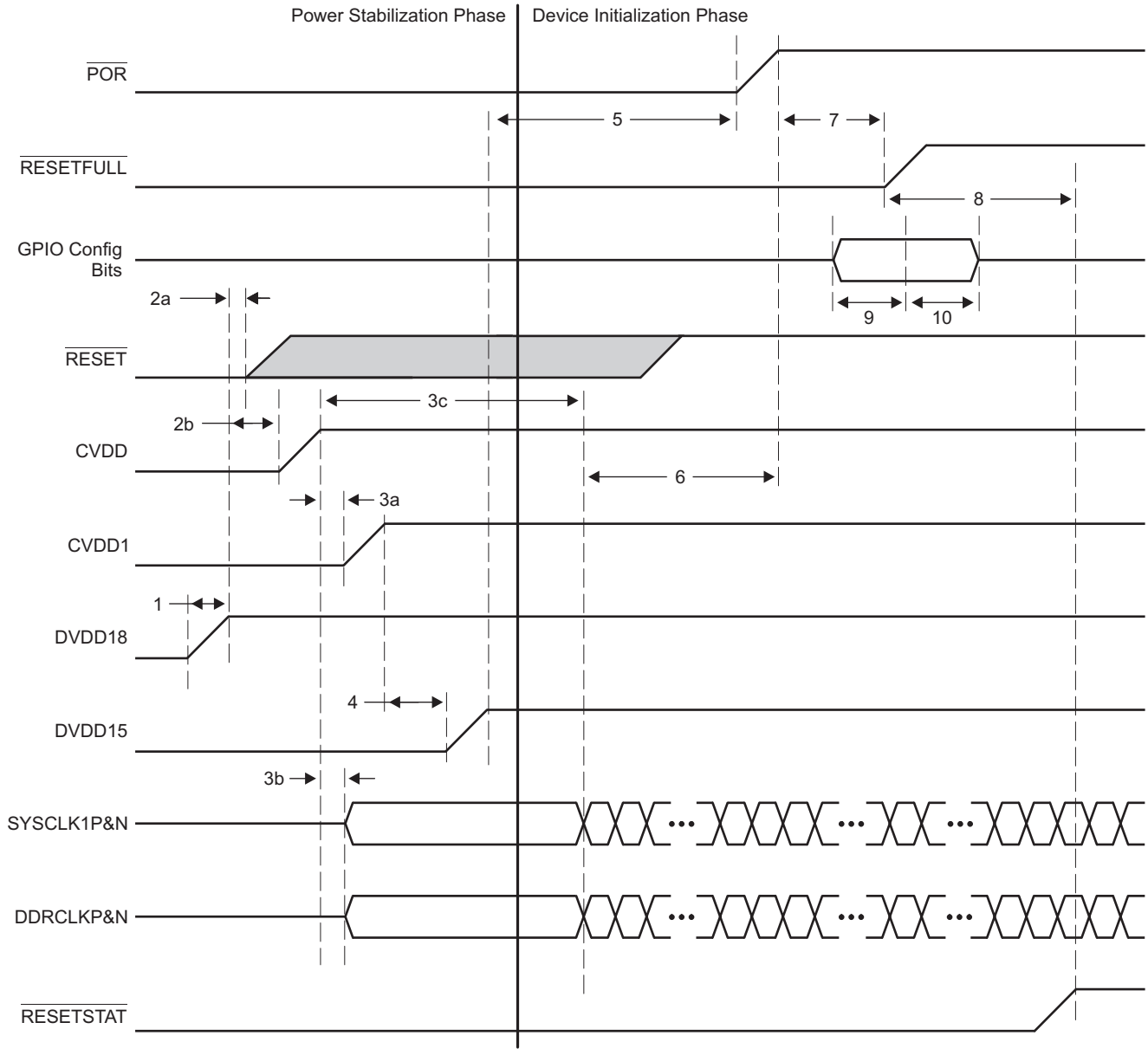


Table 7-3 IO Before Core Power Sequencing

Time	System State
1	**Begin Power Stabilization Phase** • Because $\overline{\text{POR}}$ is low, all the core logic having async reset (created from POR) are put into reset state once the core supply ramps. $\overline{\text{POR}}$ must remain low through power stabilization phase. • Filtered versions of 1.8 V can ramp simultaneously with DVDD18. • $\overline{\text{RESETSTAT}}$ is driven low once the DVDD18 supply is available. • All input and bidirectional pins must not be driven or pulled high until DVDD18 is present. Driving an input or bidirectional pin before DVDD18 could cause damage to the device.
2a	• $\overline{\text{RESET}}$ may be driven high anytime after DVDD18 is at a valid level.
2b	• CVDD (core AVS) ramps up.
3a	• CVDD1 (core constant) ramps at the same time or following CVDD. Although ramping CVDD1 and CVDD simultaneously is permitted, the voltage for CVDD1 ***must never exceed*** CVDD until after CVDD has reached a valid voltage. • The purpose of ramping up the core supplies close to each other is to reduce crowbar current. CVDD1 should trail CVDD as this will ensure that the WLs in the memories are turned off and there is no current through the memory bit cells. If, however, CVDD1 (core constant) ramps up before CVDD (core AVS), then the worst case current could be on the order of twice the specified draw of CVDD1.
3b	• Once CVDD is valid, the clock drivers should be enabled. Although the clock inputs are not necessary at this time, they should either be driven with a valid clock or held in a static state with one leg high and one leg low.
3c	• The DDRCLK and SYSCLK1 may begin to toggle anytime between when CVDD is at a valid level and the setup time before $\overline{\text{POR}}$ goes high specified by t6.
4	• DVDD15 (1.5 V) supply is ramped up following CVDD1.
5	• $\overline{\text{POR}}$ must continue to remain low for at least 100 µs after power has stabilized. End power stabilization phase
6	Begin Device Initialization • Device initialization requires 500 SYSCLK1 periods after the power stabilization phase. The maximum clock period is 33.33 nsec so a delay of an additional 16 µs is required before a rising edge of $\overline{\text{POR}}$. The clock must be active during the entire 16 µs. • $\overline{\text{POR}}$ must remain low.
7	• $\overline{\text{RESETFULL}}$ is held low for at least 24 transitions of the SYSCLK1 after $\overline{\text{POR}}$ has stabilized at a high level. • The rising edge of the $\overline{\text{RESETFULL}}$ will remove the reset to the efuse farm allowing the scan to begin.
8	• Once device initialization and the efuse farm scan are complete, the $\overline{\text{RESETSTAT}}$ signal is driven high. This delay will be 10000 to 50000 clock cycles. End device initialization phase
9	• GPIO configuration bits must be valid for at least 12 transitions of the SYSCLK1 before the rising edge of $\overline{\text{RESETFULL}}$
10	• GPIO configuration bits must be held valid for at least 12 transitions of the SYSCLK1 after the rising edge of $\overline{\text{RESETFULL}}$
End of Table 7-3	

7.2.1.3 Prolonged Resets

Holding the device in $\overline{\text{POR}}$, $\overline{\text{RESETFULL}}$, or $\overline{\text{RESET}}$ for long periods of time will affect the long-term reliability of the part. The device should not be held in a reset for times exceeding one hour at a time and no more than 5% of the total lifetime for which the device is powered-up. Exceeding these limits will cause a gradual reduction in the reliability of the part. This can be avoided by allowing the DSP to boot and then configuring it to enter a hibernation state soon after power is applied. This will satisfy the reset requirement while limiting the power consumption of the device.

7.2.1.4 Clocking During Power Sequencing

Some of the clock inputs are required to be present for the device to initialize correctly, but behavior of many of the clocks is contingent on the state of the boot configuration pins. Table 7-4 describes the clock sequencing and the conditions that affect the clock operation. Note that all clock drivers should be in a high-impedance state until CVDD is at a valid level and that all clock inputs either be active or in a static state with one leg pulled low and the other connected to CVDD.

Table 7-4 Clock Sequencing

Clock	Condition	Sequencing
DDRCLK	None	Must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
SYSCLK	CORECLKSEL = 0	SYSCLK used to clock the core PLL. It must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
	CORECLKSEL = 1	SYSCLK used only for AIF. Clock must be present before the reset to the AIF is removed.
ALTCORECLK	CORECLKSEL = 0	ALTCORECLK is not used and should be tied to a static state.
	CORECLKSEL = 1	ALTCORECLK is used to clock the core PLL. It must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
PASSCLK	PASSCLKSEL = 0	PASSCLK is not used and should be tied to a static state.
	PASSCLKSEL = 1	PASSCLK is used as a source for the PASS PLL. It must be present before the PASS PLL is removed from reset and programmed.
SRIOSGMIICLK	An SGMII port will be used.	SRIOSGMIICLK must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
	SGMII will not be used. SRIO will be used as a boot device.	SRIOSGMIICLK must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
	SGMII will not be used. SRIO will be used after boot.	SRIOSGMIICLK is used as a source to the SRIO SerDes PLL. It must be present before the SRIO is removed from reset and programmed.
	SGMII will not be used. SRIO will not be used.	SRIOSGMIICLK is not used and should be tied to a static state.
PCIECLK	PCIE will be used as a boot device.	PCIECLK must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
	PCIE will be used after boot.	PCIECLK is used as a source to the PCIE SerDes PLL. It must be present before the PCIe is removed from reset and programmed.
	PCIE will not be used.	PCIECLK is not used and should be tied to a static state.
MCMCLK	HyperLink will be used as a boot device.	MCMCLK must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
	HyperLink will be used after boot.	MCMCLK is used as a source to the HyperLink SerDes PLL. It must be present before the HyperLink is removed from reset and programmed.
	HyperLink will not be used.	MCMCLK is not used and should be tied to a static state.
End of Table 7-4		

7.2.2 Power-Down Sequence

The power down sequence is the exact reverse of the power-up sequence described above. The goal is to prevent a large amount of static current and to prevent overstress of the device. A power-good circuit that monitors all the supplies for the device should be used in all designs. If a catastrophic power supply failure occurs on any voltage rail, $\overline{\text{POR}}$ should transition to low to prevent over-current conditions that could possibly impact device reliability.

A system power monitoring solution is needed to shut down power to the board if a power supply fails. Long-term exposure to an environment in which one of the power supply voltages is no longer present will affect the reliability of the device. Holding the device in reset is not an acceptable solution because prolonged periods of time with an active reset can also affect long term reliability.

7.2.3 Power Supply Decoupling and Bulk Capacitors

In order to properly decouple the supply planes on the PCB from system noise, decoupling and bulk capacitors are required. Bulk capacitors are used to minimize the effects of low frequency current transients and decoupling or bypass capacitors are used to minimize higher frequency noise. For recommendations on selection of power supply decoupling and bulk capacitors see the *Hardware Design Guide for KeyStone Devices* in 2.9 "Related Documentation from Texas Instruments" on page 66.

7.2.4 SmartReflex

Increasing the device complexity increases its power consumption and with the smaller transistor structures responsible for higher achievable clock rates and increased performance, comes an inevitable penalty: increasing leakage currents. Leakage currents are present in any active circuit, independent of clock rates and usage scenarios. This static power consumption is mainly determined by transistor type and process technology. Higher clock rates also increase dynamic power, the power used when transistors switch. The dynamic power depends mainly on a specific usage scenario, clock rates, and I/O activity.

Texas Instruments SmartReflex technology is used to decrease both static and dynamic power consumption while maintaining the device performance. SmartReflex in the TMS320C6670 device is a feature that allows the core voltage to be optimized based on the process corner of the device. This requires a voltage regulator for each TMS320C6670 device.

To guarantee maximizing performance and minimizing power consumption of the device, SmartReflex is required to be implemented whenever the TMS320C6670 device is used. The voltage selection is done using 4 VCNTL pins which are used to select the output voltage of the core voltage regulator.

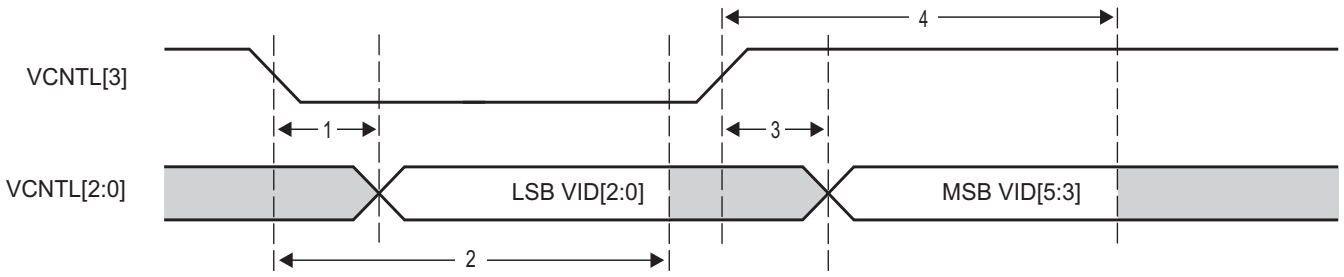
For information on implementation of SmartReflex see the *DSP Power Consumption Summary for KeyStone Devices Application Report* and the *Hardware Design Guide for KeyStone Devices* in 2.9 "Related Documentation from Texas Instruments" on page 66.

Table 7-5 SmartReflex 4-Pin VID Interface Switching Characteristics (see Figure 7-3)

No.	Parameter		Min	Max	Unit
1	td(VCNTL[2:0]-VCNTL[3])	Delay time - VCNTL[2:0] valid after VCNTL[3] low		300.00	ns
2	toh(VCNTL[3]-VCNTL[2:0])	Output hold time - VCNTL[2:0] valid after VCNTL[3]	0.07	172020C (1)	ms
3	td(VCNTL[2:0]-VCNTL[3])	Delay time - VCNTL[2:0] valid after VCNTL[3] high		300.00	ns
4	toh(VCNTL[3]-VCNTL[2:0)	Output hold time - VCNTL[2:0] valid after VCNTL[3] high	0.07	172020C	ms
End of Table 7-5					

1 C = 1/SYSCLK1 frequency (See Figure 7-9)in ms

Figure 7-3 SmartReflex 4-Pin VID Interface Timing



7.3 Power Sleep Controller (PSC)

The Power Sleep Controller (PSC) controls overall device power by turning off unused power domains and gating off clocks to individual peripherals and modules. The PSC provides the user with an interface to control several important power and clock operations.

For information on the Power Sleep Controller, see the *Power Sleep Controller (PSC) for KeyStone Devices* User Guide in 2.9 "Related Documentation from Texas Instruments" on page 66.

7.3.1 Power Domains

The device has several power domains that can be turned on for operation or off to minimize power dissipation. The global power/sleep controller (GPSC) is used to control the power gating of various power domains.

Table 7-6 shows the TMS320C6670 power domains.

Table 7-6 Power Domains

Domain	Block(s)	Note	Power Connection
0	Most peripheral logic	Cannot be disabled	Always on
1	Per-core TETB and system TETB	RAMs can be powered down	Software control
2	Network Coprocessor	Logic can be powered down	Software control
3	PCIe	Logic can be powered down	Software control
4	SRIO	Logic can be powered down	Software control
5	HyperLink	Logic can be powered down	Software control
6	Reserved	Reserved	Reserved
7	MSMC RAM	MSMC RAM can be powered down	Software control
8	RAC_A, RAC_B, and TAC	Logic can be powered down	Software control
9	FFTC_A and FFTC_B	Logic can be powered down	Software control
10	AIF2	RAMs can be powered down	Software control
11	TCP3d_A	RAMs can be powered down	Software control
12	VCP2_B, VCP2_C, and VCP2_D	RAMs can be powered down	Software control
13	C66x Core 0, L1/L2 RAMs	L2 RAMs can sleep	Software control via C66x CorePac. For details, see the C66x CorePac Reference Guide.
14	C66x Core 1, L1/L2 RAMs	L2 RAMs can sleep	
15	C66x Core 2, L1/L2 RAMs	L2 RAMs can sleep	
16	C66x Core 3, L1/L2 RAMs	L2 RAMs can sleep	
17	TCP3d_B	RAMs can be powered down	Software control
18	BCP, FFTC_C, and TCP3d_C	Logic can be powered down for BCP, FFTC_C, and RAMs can be powered down for TCP3d_C	Software control
End of Table 7-6			

7.3.2 Clock Domains

Clock gating to each logic block is managed by the local power/sleep controllers (LPSCs) of each module. For modules with a dedicated clock or multiple clocks, the LPSC communicates with the PLL controller to enable and disable that module's clock(s) at the source. For modules that share a clock with other modules, the LPSC controls the clock gating.

Table 7-7 shows the TMS320C6670 clock domains.

Table 7-7 Clock Domains

LPSC Number	Module(s)	Notes
0	Shared LPSC for all peripherals other than those listed in this table	Always on
1	SmartReflex	Always on
2	DDR3 EMIF	Always on
3	TCP3e	Software control
4	VCP2_A	Software control
5	Debug subsystem and tracers	Software control
6	Per-core TETB and system TETB	Software control
7	Packet Accelerator	Software control
8	Ethernet SGMIIs	Software control
9	Security Accelerator	Software control
10	PCIe	Software control
11	SRIO	Software control
12	HyperLink	Software control
13	Reserved	Reserved
14	MSMC RAM	Software control
15	RAC_A and RAC_B	Software control
16	TAC	Software control
17	FFTC_A and FFTC_B	Software control
18	AIF2	Software control
19	TCP3d_A	Software control
20	VCP2_B	Software control
21	VCP2_C	Software control
22	VCP2_D	Software control
23	C66x CorePac0 and Timer0	Always on
24	C66x CorePac1 and Timer1	Always on
25	C66x CorePac1 RSAs	Software control
26	C66x CorePac2 and Timer2	Always on
27	C66x CorePac2 RSAs	Software control
28	C66x CorePac3 and Timer3	Always on
29	TCP3d_B	Software control
30	BCP, FFTC_C, and TCP3d_C	Software control
No LPSC	Bootcfg, PSC, and PLL Controller	These modules do not use LPSC
End of Table 7-7		

7.3.3 PSC Register Memory Map

Table 7-8 shows the PSC Register memory map.

Table 7-8 PSC Register Memory Map (Part 1 of 3)

Offset	Register	Description
0x000	PID	Peripheral Identification Register
0x004 - 0x010	Reserved	Reserved
0x014	VCNTLID	Voltage Control Identification Register
0x018 - 0x11C	Reserved	Reserved
0x120	PTCMD	Power Domain Transition Command Register
0x124	Reserved	Reserved
0x128	PTSTAT	Power Domain Transition Status Register
0x12C - 0x1FC	Reserved	Reserved
0x200	PDSTAT0	Power Domain Status Register 0 (always on)
0x204	PDSTAT1	Power Domain Status Register 1 (Per-CorePac TETB and System TETB)
0x208	PDSTAT2	Power Domain Status Register 2 (Network Coprocessor)
0x20C	PDSTAT3	Power Domain Status Register 3 (PCIe)
0x210	PDSTAT4	Power Domain Status Register 4 (SRIO)
0x214	PDSTAT5	Power Domain Status Register 5 (HyperLink)
0x218	PDSTAT6	Power Domain Status Register 6 (Reserved)
0x21C	PDSTAT7	Power Domain Status Register 7 (MSMC RAM)
0x220	PDSTAT8	Power Domain Status Register 8 (RAC_A, RAC_B, and TAC)
0x224	PDSTAT9	Power Domain Status Register 9 (FFTC_A and FFTC_B)
0x228	PDSTAT10	Power Domain Status Register 10 (AIF2)
0x22C	PDSTAT11	Power Domain Status Register 11 (TCP3d_A)
0x230	PDSTAT12	Power Domain Status Register 12 (VCP2_B, VCP2_C and VCP2_D)
0x234	PDSTAT13	Power Domain Status Register 13 (C66x CorePac0)
0x238	PDSTAT14	Power Domain Status Register 14 (C66x CorePac1)
0x23C	PDSTAT15	Power Domain Status Register 15 (C66x CorePac2)
0x240	PDSTAT16	Power Domain Status Register 16 (C66x CorePac3)
0x244	PDSTAT17	Power Domain Status Register 17 (TCP3d_B)
0x248	PDSTAT18	Power Domain Status Register 18 (BCP, FFTC_C and TCP3d_C)
0x24C - 0x2FC	Reserved	Reserved
0x300	PDCTL0	Power Domain Control Register 0 (always on)
0x304	PDCTL1	Power Domain Control Register 1 (Per-CorePac TETB and system TETB)
0x308	PDCTL2	Power Domain Control Register 2 (Network Coprocessor)
0x30C	PDCTL3	Power Domain Control Register 3 (PCIe)
0x310	PDCTL4	Power Domain Control Register 4 (SRIO)
0x314	PDCTL5	Power Domain Control Register 5 (HyperLink)
0x318	PDCTL6	Power Domain Control Register 6 (Reserved)
0x31C	PDCTL7	Power Domain Control Register 7 (MSMC RAM)
0x320	PDCTL8	Power Domain Control Register 8 (RAC_A, RAC_B and TAC)
0x324	PDCTL9	Power Domain Control Register 9 (FFTC_A and FFTC_B)
0x328	PDCTL10	Power Domain Control Register 10 (AIF2)
0x32C	PDCTL11	Power Domain Control Register 11 (TCP3d_A)
0x330	PDCTL12	Power Domain Control Register 12 (VCP2_B, VCP2_C and VCP2_D)

Table 7-8 PSC Register Memory Map (Part 2 of 3)

Offset	Register	Description
0x334	PDCTL13	Power Domain Control Register 13 (C66x CorePac0)
0x338	PDCTL14	Power Domain Control Register 14 (C66x CorePac1)
0x33C	PDCTL15	Power Domain Control Register 15 (C66x CorePac2)
0x340	PDCTL16	Power Domain Control Register 16 (C66x CorePac3)
0x344	PDCTL17	Power Domain Control Register 17 (TCP3d_B)
0x348	PDCTL18	Power Domain Control Register 18 (BCP, FFTC_C and TCP3d_C)
0x34C - 0x7FC	Reserved	Reserved
0x800	MDSTAT0	Module Status Register 0 (never gated)
0x804	MDSTAT1	Module Status Register 1 (SmartReflex)
0x808	MDSTAT2	Module Status Register 2 (DDR3 EMIF)
0x80C	MDSTAT3	Module Status Register 3 (TCP3e)
0x810	MDSTAT4	Module Status Register 4 (VCP2_A)
0x814	MDSTAT5	Module Status Register 5 (debug subsystem and tracers)
0x818	MDSTAT6	Module Status Register 6 (per-CorePac TETB and system TETB)
0x81C	MDSTAT7	Module Status Register 7 (Packet Accelerator)
0x820	MDSTAT8	Module Status Register 8 (Ethernet SGMIIs)
0x824	MDSTAT9	Module Status Register 9 (Security Accelerator)
0x828	MDSTAT10	Module Status Register 10 (PCIe)
0x82C	MDSTAT11	Module Status Register 11 (SRIO)
0x830	MDSTAT12	Module Status Register 12 (HyperLink)
0x834	MDSTAT13	Module Status Register 13 (Reserved)
0x838	MDSTAT14	Module Status Register 14 (MSMC RAM)
0x83C	MDSTAT15	Module Status Register 15 (RAC_A and RAC_B)
0x840	MDSTAT16	Module Status Register 16 (TAC)
0x844	MDSTAT17	Module Status Register 17 (FFTC_A and FFTC_B)
0x848	MDSTAT18	Module Status Register 18 (AIF2)
0x84C	MDSTAT19	Module Status Register 19 (TCP3d_A)
0x850	MDSTAT20	Module Status Register 20 (VCP2_B)
0x854	MDSTAT21	Module Status Register 21 (VCP2_C)
0x858	MDSTAT22	Module Status Register 22 (VCP2_D)
0x85C	MDSTAT23	Module Status Register 23 (C66x CorePac0 and Timer 0)
0x860	MDSTAT24	Module Status Register 24 (C66x CorePac1 and Timer 1)
0x864	MDSTAT25	Module Status Register 25 (C66x CorePac1 RSAs)
0x868	MDSTAT26	Module Status Register 26 (C66x CorePac2 and Timer 2)
0x86C	MDSTAT27	Module Status Register 27 (C66x CorePac2 RSAs)
0x870	MDSTAT28	Module Status Register 28 (C66x CorePac3 and Timer 3)
0x874	MDSTAT29	Module Status Register 29 (TCP3d_B)
0x878	MDSTAT30	Module Status Register 30 (BCP, FFTC_C and TCP3d_C)
0x87C - 0x9FC	Reserved	Reserved
0xA00	MDCTL0	Module Control Register 0 (never gated)
0xA04	MDCTL1	Module Control Register 1 (SmartReflex)
0xA08	MDCTL2	Module Control Register 2 (DDR3 EMIF)
0xA0C	MDCTL3	Module Control Register 3 (TCP3e)
0xA10	MDCTL4	Module Control Register 4 (VCP2_A)

Table 7-8 PSC Register Memory Map (Part 3 of 3)

Offset	Register	Description
0xA14	MDCTL5	Module Control Register 5 (debug subsystem and tracers)
0xA18	MDCTL6	Module Control Register 6 (per-CorePac TETB and system TETB)
0xA1C	MDCTL7	Module Control Register 7 (Packet Accelerator)
0xA20	MDCTL8	Module Control Register 8 (Ethernet SGMIIs)
0xA24	MDCTL9	Module Control Register 9 (Security Accelerator)
0xA28	MDCTL10	Module Control Register 10 (PCIe)
0xA2C	MDCTL11	Module Control Register 11 (SRIO)
0xA30	MDCTL12	Module Control Register 12 (HyperLink)
0xA34	MDCTL13	Module Control Register 13 (Reserved)
0xA38	MDCTL14	Module Control Register 14 (MSMC RAM)
0xA3C	MDCTL15	Module Control Register 15 (RAC_A and RAC_B)
0xA40	MDCTL16	Module Control Register 16 (TAC)
0xA44	MDCTL17	Module Control Register 17 (FFTC_A and FFTC_B)
0xA48	MDCTL18	Module Control Register 18 (AIF2)
0xA4C	MDCTL19	Module Control Register 19 (TCP3d_A)
0xA50	MDCTL20	Module Control Register 20 (VCP2_B)
0xA54	MDCTL21	Module Control Register 21 (VCP2_C)
0xA58	MDCTL22	Module Control Register 22 (VCP2_D)
0xA5C	MDCTL23	Module Control Register 23 (C66x CorePac0 and Timer 0)
0xA60	MDCTL24	Module Control Register 24 (C66x CorePac1and Timer 1)
0xA64	MDCTL25	Module Control Register 25 (C66x CorePac1 RSAs)
0xA68	MDCTL26	Module Control Register 26 (C66x Corepac2 and Timer 2)
0xA6C	MDCTL27	Module Control Register 27 (C66x CorePac2 RSAs)
0xA70	MDCTL28	Module Control Register 28 (C66x CorePac3 and Timer 3)
0xA74	MDCTL29	Module Control Register 29 (TCP3d_B)
0xA78	MDCTL30	Module Control Register 30(BCP, FFTC_C and TCP3d_C)
0xA7C - 0xFFC	Reserved	Reserved
End of Table 7-8		

7.4 Reset Controller

The reset controller detects the different type of resets supported on the TMS320C6670 device and manages the distribution of those resets throughout the device.

The device has the following types of resets:

- Power-on reset
- Hard reset
- Soft reset
- Local reset

Table 7-9 explains further the types of reset, the reset initiator, and the effects of each reset on the device. For more information on the effects of each reset on the PLL controllers and their clocks, see Section 7.4.7 "Reset Electrical Data/Timing" on page 126.

Table 7-9 Reset Types

Type	Initiator	Effect(s)
Power-on Reset	$\overline{\text{POR}}$ pin $\overline{\text{RESETFULL}}$ pin	Resets the entire chip including the test and emulation logic. The device configuration pins are latched only during power-on reset.
Hard Reset	$\overline{\text{RESET}}$ pin PLLCTL [1] register (RSCTRL) Watchdog timers Emulation	Hard reset resets everything except for test, emulation logic and reset isolation modules. This reset is also different from power-on reset in that the PLLCTL assumes power and clocks are stable when hard reset is asserted. The device configurations pins are not re-latched. Emulation initiated reset is always a hard reset. By default these initiators are configured as Hard reset, but can be configured (Except Emulation) as soft reset in the RSCFG register of PLLCTL. Contents of DDR3 SDRAM memory can be retained during a hard reset if the SDRAM is placed in self-refresh mode.
Soft Reset	$\overline{\text{RESET}}$ pin PLLCTL register (RSCTRL) Watchdog timers	Soft Reset will behave like hard reset except that PCIe MMRs (memory-mapped registers) and DDR3 EMIF MMRs contents are retained. By default these initiators are configured as hard reset, but can be configured as Soft reset in the RSCFG register of PLLCTL. Contents of DDR3 SDRAM memory can be retained during a soft reset if the SDRAM is placed in self-refresh mode.
Local Reset	$\overline{\text{LRESET}}$ pin Watchdog timer timeout LPSC MMRs	Resets the CorePac, without disturbing clock alignment or memory contents. The device configuration pins are not re-latched.
End of Table 7-9		

1 All masters in the device have access to the PLLCTL registers.

7.4.1 Power-on Reset

Power-on reset is used to reset the entire device, including the test and emulation logic.

Power-on reset is initiated by the following

1. $\overline{\text{POR}}$ pin
2. $\overline{\text{RESETFULL}}$ pin

During power-up, the $\overline{\text{POR}}$ pin must be asserted (driven low) until the power supplies have reached their normal operating conditions. Also a $\overline{\text{RESETFULL}}$ pin is provided to allow reset of the entire device, including the reset-isolated logic, when the device is already powered up. For this reason, the $\overline{\text{RESETFULL}}$ pin, unlike $\overline{\text{POR}}$, should be driven by the on-board host control other than the power good circuitry. For power-on reset, the Main PLL Controller comes up in bypass mode and the PLL is not enabled. Other resets do not affect the state of the PLL or the dividers in the PLL Controller.

The following sequence must be followed during a power-on reset:

1. Wait for all power supplies to reach normal operating conditions while keeping the POR pin asserted (driven low). While POR is asserted, all pins except RESETSTAT will be set to high-impedance. After the POR pin is de-asserted (driven high), all Z group pins, low group pins, and high group pins are set to their reset state and will remain at their reset state until otherwise configured by their respective peripheral. All peripherals that are power managed, are disabled after a power-on reset and must be enabled through the Device State Control Registers (for more details, see Section Table 3-2 "Device State Control Registers" on page 68).
2. Clocks are reset, and they are propagated throughout the chip to reset any logic that was using reset synchronously. All logic is now reset and RESETSTAT will be driven low, indicating that the device is in reset.
3. POR must be held active until all supplies on the board are stable, and then for at least an additional period of time (as specified in Section 7.2.1 "Power-Up Sequencing" on page 110) for the chip-level PLLs to lock.
4. The POR pin can now be de-asserted. Reset-sampled pin values are latched at this point. Then all chip-level PLLs are taken out of reset, they begin their locking sequence, and all power-on device initialization processes begin.
5. After device initialization is complete, the RESETSTAT pin is de-asserted (driven high). By this time, the DDR3 PLL has already completed its locking sequence and is supplying a valid clock. The system clocks of both PLL controllers are allowed to finish their current cycles and then paused for 10 cycles of their respective system reference clocks. After the pause, the system clocks are restarted at their default divide by settings.
6. The device is now out of reset and device execution begins as dictated by the selected boot mode.

Note—To most of the device, reset is de-asserted only when the POR and RESET pins are both de-asserted (driven high). Therefore, in the sequence described above, if the RESET pin is held low past the low period of the POR pin, most of the device will remain in reset. The RESET pin should not be tied to the POR pin.

7.4.2 Hard Reset

A hard reset will reset everything on the device except the PLLs, test, emulation logic, and reset-isolated modules. POR should also remain de-asserted during this time.

Hard reset is initiated by the following:

- RESET pin
- RSCTRL Register in PLLCTL
- Watchdog timer
- Emulation

All the above initiators, by default, are configured to act as hard reset. Except emulation, all of the other 3 initiators can be configured as soft resets in the RSCFG Register in PLLCTL.

The following sequence must be followed during a hard reset:

1. The RESET pin is pulled active low for a minimum of 24 CLKIN1 cycles. During this time the RESET signal is able to propagate to all modules (except those specifically mentioned above). All I/O are Hi-Z for modules affected by RESET, to prevent off-chip contention during the warm reset.
2. Once all logic is reset, RESETSTAT is driven active to denote that the device is in reset.
3. The RESET pin can now be released. A minimal device initialization begins to occur. Note that configuration pins are not re-latched and clocking is unaffected within the device.
4. After device initialization is complete, the RESETSTAT pin is de-asserted (driven high).

Note—The POR pin should be held inactive (high) throughout the warm reset sequence. Otherwise, if POR is activated (brought low), the minimum POR pulse width must be met. The RESET pin should not be tied to the POR pin.

7.4.3 Soft Reset

A soft reset will behave like a hard reset except that the PCIe MMRs (memory-mapped registers) and DDR3 EMIF MMRs contents are retained. $\overline{\text{POR}}$ should also remain de-asserted during this time.

Soft reset is initiated by the following

- $\overline{\text{RESET}}$ pin
- RSCTRL Register in PLLCTL
- Watchdog timer

All the above initiators by default are configured to act as hard reset. Except emulation, all of the other 3 initiators can be configured as soft resets in the RSCFG register in PLLCTL.

In the case of a soft reset, the clock logic or the power control logic of the peripherals is not affected, and, therefore, the enabled/disabled state of the peripherals is not affected. On a soft reset, the DDR3 memory controller registers are ***not*** reset. In addition, the DDR3 SDRAM memory content is retained if the user places the DDR3 SDRAM in self-refresh mode before invoking the soft reset.

During a soft reset, the following happens:

1. The $\overline{\text{RESETSTAT}}$ pin goes low to indicate an internal reset is being generated. The reset is allowed to propagate through the system. Internal system clocks are not affected. PLLs also remain locked.
2. After device initialization is complete, the $\overline{\text{RESETSTAT}}$ pin is deasserted (driven high). In addition, the PLL Controllers pause their system clocks for about 8 cycles.

   At this point:

   - The state of the peripherals before the soft reset is not changed.
   - The I/O pins are controlled as dictated by the DEVSTAT register.
   - The DDR3 MMRs and PCIe MMRs retain their previous values. Only the DDR3 memory controller and PCIe state machines are reset by the soft reset.
   - The PLL controllers are operating in the mode prior to soft reset. System clocks are unaffected.

The boot sequence is started after the system clocks are restarted. Because the configuration pins are not latched with a system reset, the previous values, as shown in the DEVSTAT Register, are used to select the boot mode.

7.4.4 Local Reset

The local reset can be used to reset a particular CorePac without resetting any other device components.

Local reset is initiated by the following (for more details see the *Phase Locked Loop (PLL) Controller for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66):

- $\overline{\text{LRESET}}$ pin
- Watchdog timer should cause one of the below based on the setting of the CORESEL[2:0] and RSTCFG registers in the PLL Controller. See "Reset Configuration Register (RSTCFG)" on page 136 and "CIC Registers" on page 170.
  - Local reset
  - NMI
  - NMI followed by a time delay and then a local reset for the CorePac selected
  - Hard reset by requesting reset via PLLCTL
- LPSC MMRs (memory-mapped registers)

7.4.5 Reset Priority

If any of the above reset sources occur simultaneously, the PLLCTL processes only the highest priority reset request. The reset request priorities are as follows (high to low):

- Power-on reset
- Hard/soft reset

7.4.6 Reset Controller Register

The reset controller register are part of the PLLCTL MMRs. All C6670 device-specific MMRs are covered in Section 7.5.2 "PLL Controller Memory Map" on page 131. For more details on these registers and how to program them, see the *Phase Locked Loop (PLL) Controller for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

7.4.7 Reset Electrical Data/Timing

Table 7-10 Reset Timing Requirements (1)
(see Figure 7-4 and Figure 7-5)

No.			Min	Max	Unit
		RESETFULL Pin Reset			
1	tw(RESETFULL)	Pulse width - pulse width RESETFULL low	500C		ns
		Soft/Hard-Reset			
2	tw(RESET)	Pulse width - pulse width RESET low	500C		ns
End of Table 7-10					

1 C = 1/SYSCLK1 clock frequency in ns

Table 7-11 Reset Switching Characteristics (1)
(see Figure 7-4 and Figure 7-5)

No.	Parameter		Min	Max	Unit
		RESETFULL Pin Reset			
3	td(RESETFULLH-RESETSTATH)	Delay time - RESETSTAT high after RESETFULL high		50000C	ns
		Soft/Hard Reset			
4	td(RESETH-RESETSTATH)	Delay time - RESETSTAT high after RESET high		50000C	ns
End of Table 7-11					

1 C = 1/SYSCLK1 clock frequency in ns

Figure 7-4 RESETFULL Reset Timing

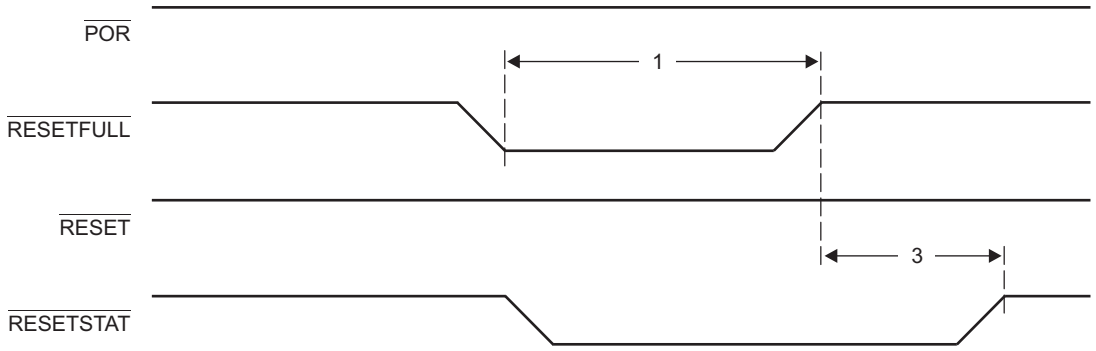


Figure 7-5 Soft/Hard Reset Timing

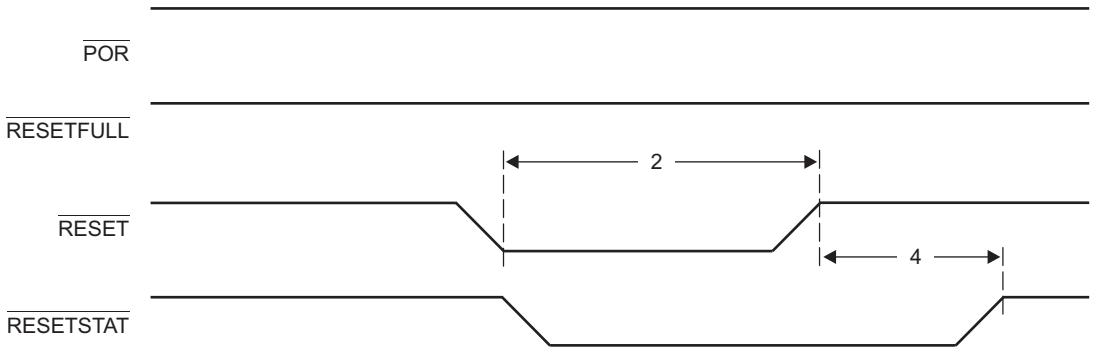
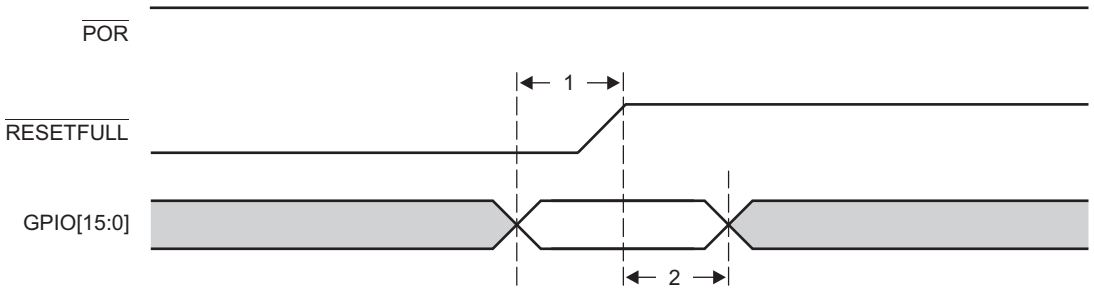


Table 7-12 Boot Configuration Timing Requirements (1)
See Figure 7-6)

No.			Min	Max	Unit
1	tsu(GPIOn-$\overline{\text{RESETFULL}}$)	Setup time - GPIO valid before $\overline{\text{RESETFULL}}$ asserted	12C		ns
2	th($\overline{\text{RESETFULL}}$-GPIOn)	Hold time - GPIO valid after $\overline{\text{RESETFULL}}$ asserted	12C		ns
End of Table 7-12					

1 C = 1/SYSCLK1 clock frequency in ns.

Figure 7-6 Boot Configuration Timing

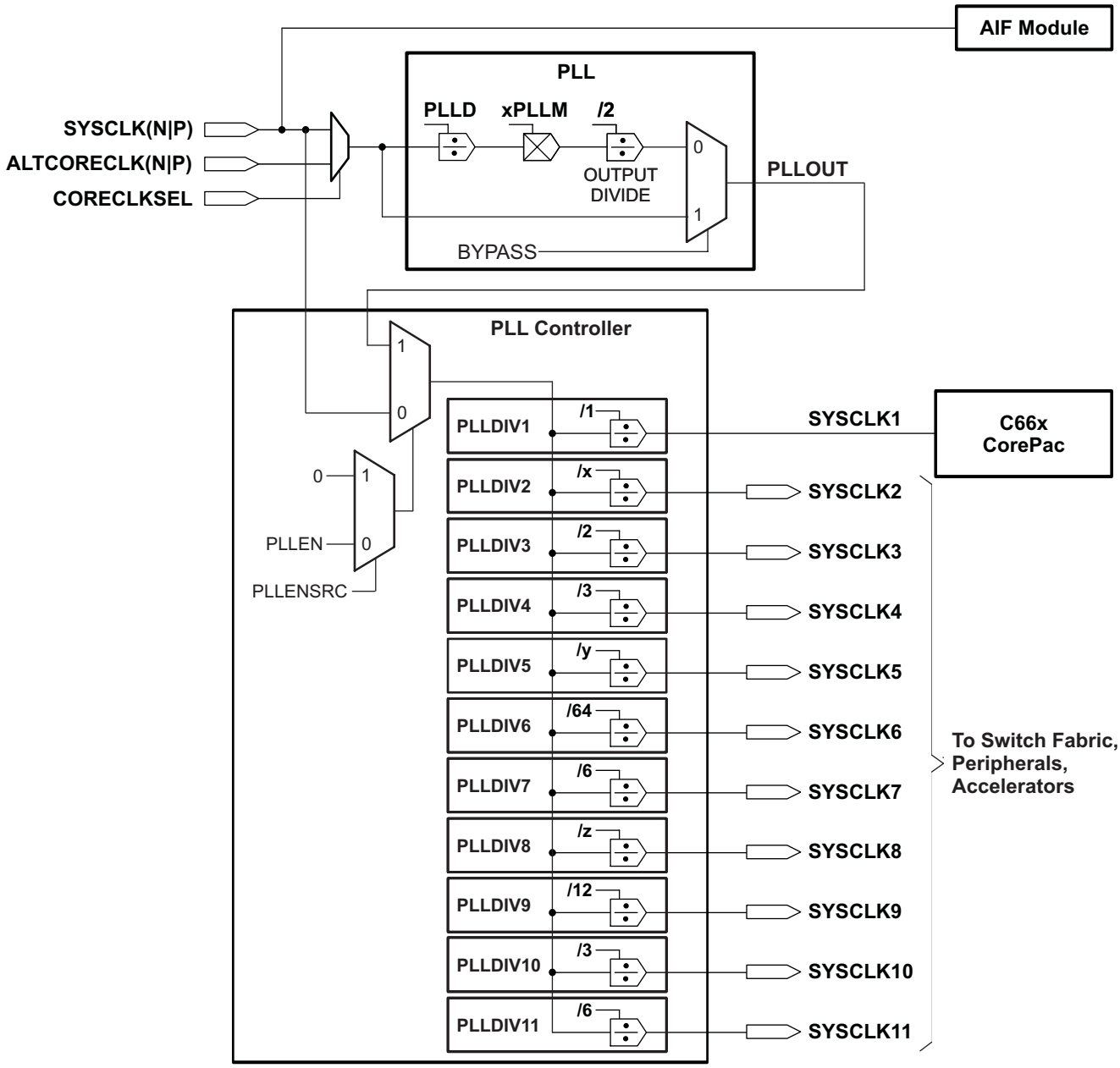


7.5 Main PLL and the PLL Controller

This section provides a description of the Main PLL and the PLL Controller. For details on the operation of the PLL Controller module, see the *Phase Locked Loop (PLL) Controller for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

The Main PLL is controlled by the standard PLL Controller. The PLL Controller manages the clock ratios, alignment, and gating for the system clocks to the device. Figure 7-7 shows a block diagram of the Main PLL and the PLL Controller.

Figure 7-7 Main PLL and PLL Controller



Note—The Main PLL Controller registers can be accessed by any master in the device. The PLLM[5:0] bits of the multiplier are controlled by the PLLM Register inside the PLL Controller and the PLLM[12:6] bits are controlled by the chip-level MAINPLLCTL0 Register. The Output Divide and Bypass logic of the PLL are controlled by fields in the SECCTL Register in the PLL Controller. Only PLLDIV2, PLLDIV5, and PLLDIV8 are programmable on the device. See the *Phase Locked Loop (PLL) Controller for KeyStone Devices User Guide* in section 2.9 "Related Documentation from Texas Instruments" on page 66 for more details on how to program the PLL controller.

The multiplication and division ratios within the PLL and the post-division for each of the chip-level clocks are determined by a combination of this PLL and the PLL Controller. The PLL Controller also controls reset propagation through the chip, clock alignment, and test points. The PLL Controller monitors the PLL status and provides an output signal indicating when the PLL is locked.

Main PLL power is supplied externally via the Main PLL power-supply pin (AVDDA1). An external EMI filter circuit must be added to all PLL supplies. See the *Hardware Design Guide for KeyStone Devices* in section 2.9 "Related Documentation from Texas Instruments" on page 66 for detailed recommendations. For the best performance, TI recommends that all the PLL external components be on a single side of the board without jumpers, switches, or components other than those shown. For reduced PLL jitter, maximize the spacing between switching signal traces and the PLL external components (C1, C2, and the EMI Filter).

The minimum SYSCLK rise and fall times should also be observed. For the input clock timing requirements, see Section 7.5.5 "Main PLL Controller/SRIO/HyperLink/PCIe Clock Input Electrical Data/Timing".

CAUTION—The PLL Controller module as described in the see the *Phase Locked Loop (PLL) Controller for KeyStone Devices User Guide* in2.9 "Related Documentation from Texas Instruments" on page 66 includes a superset of features, some of which are not supported on the TMS320C6670 device. The following sections describe the registers that are supported; it should be assumed that any registers not included in these sections is not supported by the device. Furthermore, only the bits within the registers described here are supported. Avoid writing to any reserved memory location or changing the value of reserved bits.

7.5.1 Main PLL Controller Device-Specific Information

7.5.1.1 Internal Clocks and Maximum Operating Frequencies

The Main PLL, used to drive the CorePacs, the switch fabric, and a majority of the peripheral clocks (all but the DDR3 and the PASS modules) requires a PLL controller to manage the various clock divisions, gating, and synchronization. The Main PLL's PLL Controller has several SYSCLK outputs that are listed below, along with the clock description. Each SYSCLK has a corresponding divider that divides down the output clock of the PLL. Note that dividers are not programmable unless explicitly mentioned in the description below.

- **SYSCLK1:** Full-rate clock for CorePac0~CorePac3, RAC, and RSA.
- **SYSCLK2:** 1/x-rate clock for CorePac (emulation) and the ADTF module. Default rate for this is 1/3. This is programmable from /1 to /32, where this clock does not violate the max of 350 MHz. SYSCLK2 can be turned off by software.
- **SYSCLK3:** 1/2-rate clock used to clock the MSMC, TCP3d, HyperLink, CPU/2 switch fabric, DDR EMIF and CPU/2 EDMA.
- **SYSCLK4:** 1/3-rate clock for the switch fabrics and fast peripherals. The Debug_SS and ETBs will use this as well.
- **SYSCLK5:** 1/y-rate clock used for system trace module. Default rate for this clock is 1/5, but is configurable to a maximum of 210 MHz and a minimum of 32 MHz. SYSCLK5 can be turned off by software.
- **SYSCLK6:** 1/64-rate clock. 1/64 rate clock (emif_ptv) used to clock the PVT-compensated buffers for DDR3 EMIF.

- **SYSCLK7:** 1/6-rate clock for slow peripherals and sources the SYSCLKOUT output pin.
- **SYSCLK8:** 1/z-rate clock. This clock is used as slow_sysclck in the system. Default for this is 1/64. This is programmable from /24 to /80.
- **SYSCLK9:** 1/12-rate clock for SmartReflex.
- **SYSCLK10:** 1/3-rate clock for SRIO only.
- **SYSCLK11:** 1/6-rate clock for PSC only.

Only SYSCLK2, SYSCLK5, and SYSCLK8 are programmable on the TMS320C6670 device.

Note—In case any of the other programmable SYSCLKs are set slower than 1/64 rate, then SYSCLK8 (SLOW_SYSCLK) needs to be programmed to either match, or be slower than, the slowest SYSCLK in the system.

7.5.1.2 Main PLL Controller Operating Modes

The Main PLL Controller has two modes of operation: bypass mode and PLL mode. The mode of operation is determined by the BYPASS bit of the PLL Secondary Control Register (SECCTL). In PLL mode, SYSCLK1 is generated from the PLL output using the values set in the PLLM and PLLD fields in the MAINPLLCTL0 Register. In bypass mode, PLL input is fed directly out as SYSCLK1.

All hosts must hold off accesses to the DSP while the frequency of its internal clocks is changing. A mechanism must be in place such that the DSP notifies the host when the PLL configuration has completed.

7.5.1.3 Main PLL Stabilization, Lock, and Reset Times

The PLL stabilization time is the amount of time that must be allotted for the internal PLL regulators to become stable after device powerup. The PLL should not be operated until this stabilization time has elapsed.

The PLL reset time is the amount of wait time needed when resetting the PLL (writing PLLRST = 1), in order for the PLL to properly reset, before bringing the PLL out of reset (writing PLLRST = 0). For the Main PLL reset time value, see Table 7-13.

The PLL lock time is the amount of time needed from when the PLL is taken out of reset (PLLRST = 1) to when to when the PLL Controller can be switched to PLL mode. The Main PLL lock time is given in Table 7-13.

Table 7-13 Main PLL Stabilization, Lock, and Reset Times

	Min	Typ	Max	Unit
PLL stabilization time	100			µs
PLL lock time			500 ×(PLLD [1]+1) × C [2]	
PLL reset time	1000			ns
End of Table 7-13				

1 PLLD is the value in PLLD bit fields of MAINPLLCTL0 register
2 C = SYSCLK1(N|P) cycle time in ns.

7.5.2 PLL Controller Memory Map

The memory map of the PLL Controller is shown in Table 7-14. TMS320C6670-specific PLL Controller Register definitions can be found in the sections following Table 7-14. For other registers in the table, see the *Phase Locked Loop (PLL) Controller for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

CAUTION—Note that only registers documented here are accessible on the TMS320C6670. Other addresses in the PLL Controller memory map including the Reserved registers should not be modified. Furthermore, only the bits within the registers described here are supported. Avoid writing to any Reserved memory location or changing the value of reserved bits. It is recommended to use read-modify-write sequence to make any changes to the valid bits in the register.

Table 7-14 PLL Controller Registers (Including Reset Controller) (Part 1 of 2)

Hex Address Range	Acronym	Register Name
0231 0000 - 0231 00E3	-	Reserved
0231 00E4	RSTYPE	Reset Type Status Register (Reset Controller)
0231 00E8	RSTCTRL	Software Reset Control Register (Reset Controller)
0231 00EC	RSTCFG	Reset Configuration Register (Reset Controller)
0231 00F0	RSISO	Reset Isolation Register (Reset Controller)
0231 00F0 - 0231 00FF	-	Reserved
0231 0100	PLLCTL	PLL Control Register
0231 0104	-	Reserved
0231 0108	SECCTL	PLL Secondary Control Register
0231 010C	-	Reserved
0231 0110	PLLM	PLL Multiplier Control Register
0231 0114	-	Reserved
0231 0118	PLLDIV1	Reserved
0231 011C	PLLDIV2	PLL Controller Divider 2 Register
0231 0120	PLLDIV3	Reserved
0231 0124	-	Reserved
0231 0128	-	Reserved
0231 012C - 0231 0134	-	Reserved
0231 0138	PLLCMD	PLL Controller Command Register
0231 013C	PLLSTAT	PLL Controller Status Register
0231 0140	ALNCTL	PLL Controller Clock Align Control Register
0231 0144	DCHANGE	PLLDIV Ratio Change Status Register
0231 0148	CKEN	Reserved
0231 014C	CKSTAT	Reserved
0231 0150	SYSTAT	SYSCLK Status Register
0231 0154 - 0231 015C	-	Reserved
0231 0160	PLLDIV4	Reserved
0231 0164	PLLDIV5	PLL Controller Divider 5 Register
0231 0168	PLLDIV6	Reserved
0231 016C	PLLDIV7	Reserved
0231 0170	PLLDIV8	PLL Controller Divider 8 Register

Table 7-14 PLL Controller Registers (Including Reset Controller) (Part 2 of 2)

Hex Address Range	Acronym	Register Name
0231 0174 - 0231 0193	PLLDIV9 - PLLDIV16	Reserved
0231 0194 - 0231 01FF	-	Reserved
End of Table 7-14		

7.5.2.1 PLL Secondary Control Register (SECCTL)

The PLL Secondary Control Register contains extra fields to control the Main PLL and is shown in Figure 7-8 and described in Table 7-15.

Figure 7-8 PLL Secondary Control Register (SECCTL))

31 24	23	22 19	18 0
Reserved	BYPASS	OUTPUT DIVIDE	Reserved
R-0000 0000	RW-1	RW-0001	RW-001 0000 0000 0000 0000

Legend: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 7-15 PLL Secondary Control Register Field Descriptions

Bit	Field	Description
31-24	Reserved	Reserved
23	BYPASS	Main PLL bypass enable 0 = Main PLL bypass disabled 1 = Main PLL bypass enabled
22-19	OUTPUT DIVIDE	Output divider ratio bits 0h = ÷1. Divide frequency by 1 1h = ÷2. Divide frequency by 2 2h - Fh = Reserved
18-0	Reserved	Reserved
End of Table 7-15		

7.5.2.2 PLL Controller Divider Register (PLLDIV2, PLLDIV5, and PLLDIV8)

The PLL Controller Divider Registers (PLLDIV2, PLLDIV5, and PLLDIV8) are shown in Figure 7-9 and described in Table 7-16. The default values of the RATIO field on a reset for PLLDIV2, PLLDIV5, and PLLDIV8 are different and mentioned in the footnote of Figure 7-9.

Figure 7-9 PLL Controller Divider Register (PLLDIVn)

31 16	15	14 8	7 0
Reserved	Dn (1) EN	Reserved	RATIO
R-0	R/W-1	R-0	R/W-n (2)

Legend: R/W = Read/Write; R = Read only; -*n* = value after reset

1 D2EN for PLLDIV2; D5EN for PLLDIV5; D8EN for PLLDIV8
2 n=02h for PLLDIV2; n=04h for PLLDIV5; n=3Fh for PLLDIV8

Table 7-16 PLL Controller Divider Register Field Descriptions

Bit	Field	Description
31-16	Reserved	Reserved
15	D*n*EN	Divider D*n* enable bit (See footnote of Figure 7-9) 0 = Divider *n* is disabled 1 = No clock output. Divider *n* is enabled.
14-8	Reserved	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
7-0	RATIO	Divider ratio bits (See footnote of Figure 7-9) 0h = ÷1. Divide frequency by 1 1h = ÷2. Divide frequency by 2 2h = ÷3. Divide frequency by 3 3h = ÷4. Divide frequency by 4 4h - 4Fh = ÷5 to ÷80. Divide frequency range: divide frequency by 5 to divide frequency by 80.
End of Table 7-16		

7.5.2.3 PLL Controller Clock Align Control Register (ALNCTL)

The PLL Controller Clock Align Control Register (ALNCTL) is shown in Figure 7-10 and described in Table 7-17.

Figure 7-10 PLL Controller Clock Align Control Register (ALNCTL)

31 8	7	6 5	4	3 2	1	0
Reserved	ALN8	Reserved	ALN5	Reserved	ALN2	Reserved
R-0	R/W-1	R-0	R/W-1	R-0	R/W-1	R-0

Legend: R/W = Read/Write; R = Read only; -*n* = value after reset, for reset value

Table 7-17 PLL Controller Clock Align Control Register Field Descriptions

Bit	Field	Description
31-8 6-5 3-2 0	Reserved	Reserved. This location is always read as 0. A value written to this field has no effect.
7 4 1	ALN8 ALN5 ALN2	SYSCLK*n* alignment. Do not change the default values of these fields. 0 = Do not align SYSCLK*n* to other SYSCLKs during GO operation. If SYS*n* in DCHANGE is set, SYSCLK*n* switches to the new ratio immediately after the GOSET bit in PLLCMD is set. 1 = Align SYSCLK*n* to other SYSCLKs selected in ALNCTL when the GOSET bit in PLLCMD is set and SYS*n* in DCHANGE is 1. The SYSCLK*n* rate is set to the ratio programmed in the RATIO bit in PLLDIV*n*.
End of Table 7-17		

7.5.2.4 PLLDIV Divider Ratio Change Status Register (DCHANGE)

Whenever a different ratio is written to the PLLDIV*n* registers, the PLLCTL flags the change in the DCHANGE Status Register. During the GO operation, the PLL controller will change only the divide ratio of the SYSCLKs with the bit set in DCHANGE. Note that the ALNCTL Register determines if that clock also needs to be aligned to other clocks. The PLLDIV Divider Ratio Change Status Register is shown in Figure 7-11 and described in Table 7-18.

Figure 7-11 PLLDIV Divider Ratio Change Status Register (DCHANGE)

31 – 8	7	6 – 5	4	3 – 2	1	0
Reserved	SYS8	Reserved	SYS5	Reserved	SYS2	Reserved
R-0	R/W-0	R-0	R/W-0	R-0	R/W-0	R-0

Legend: R/W = Read/Write; R = Read only; -*n* = value after reset, for reset value

Table 7-18 PLLDIV Divider Ratio Change Status Register Field Descriptions

Bit	Field	Description
31-8 6-5 3-2 0	Reserved	Reserved. This bit location is always read as 0. A value written to this field has no effect.
7 4 1	SYS8 SYS5 SYS2	Identifies when the SYSCLK*n* divide ratio has been modified. 0 = SYSCLK*n* ratio has not been modified. When GOSET is set, SYSCLK*n* will not be affected. 1 = SYSCLK*n* ratio has been modified. When GOSET is set, SYSCLK*n* will change to the new ratio.
End of Table 7-18		

7.5.2.5 SYSCLK Status Register (SYSTAT)

The SYSCLK Status Register (SYSTAT) shows the status of SYSCLK[11:1]. SYSTAT is shown in Figure 7-12 and described in Table 7-19.

Figure 7-12 SYSCLK Status Register (SYSTAT)

31 – 11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SYS11ON	SYS10ON	SYS9ON	SYS8ON	SYS7ON	SYS6ON	SYS5ON	SYS4ON	SYS3ON	SYS2ON	SYS1ON
R-n	R-1	R-1	R-1	R-1	R-1	R-1	R-1	R-1	R-1	R-1	R-1

Legend: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 7-19 SYSCLK Status Register Field Descriptions

Bit	Field	Description
31-11	Reserved	Reserved. This location is always read as 0. A value written to this field has no effect.
10-0	SYS[N [(1)]]ON	SYSCLK[N] on status 0 = SYSCLK[N] is gated 1 = SYSCLK[N] is on
End of Table 7-19		

1 Where N = 1, 2, 3,....N (Not all these output clocks may be used on a specific device. For more information, see the device-specific data manual)

7.5.2.6 Reset Type Status Register (RSTYPE)

The Reset Type Status (RSTYPE) Register latches the cause of the last reset. If multiple reset sources occur simultaneously, this register latches the highest priority reset source. The Reset Type Status Register is shown in Figure 7-13 and described in Table 7-20.

Figure 7-13 Reset Type Status Register (RSTYPE)

31 29	28	27 12	11 8	7 3	2	1	0
Reserved	EMU-RST	Reserved	WDRST[N]	Reserved	PLLCTRLRST	$\overline{\text{RESET}}$	POR
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

Legend: R = Read only; -*n* = value after reset

Table 7-20 Reset Type Status Register Field Descriptions

Bit	Field	Description
31-29	Reserved	Reserved. Read only. Always reads as 0. Writes have no effect.
28	EMU-RST	Reset initiated by emulation 0 = Not the last reset to occur 1 = The last reset to occur
27-12	Reserved	Reserved. Read only. Always reads as 0. Writes have no effect.
11 10 9 8	WDRST3 WDRST2 WDRST1 WDRST0	Reset initiated by Watchdog Timer[N] 0 = Not the last reset to occur 1 = The last reset to occur
7-3	Reserved	Reserved. Read only. Always reads as 0. Writes have no effect.
2	PLLCTLRST	Reset initiated by PLLCTL 0 = Not the last reset to occur 1 = The last reset to occur
1	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$ reset 0 = $\overline{\text{RESET}}$ was not the last reset to occur 1 = $\overline{\text{RESET}}$ was the last reset to occur
0	POR	Power-on reset 0 = Power-on reset was not the last reset to occur 1 = Power-on reset was the last reset to occur
End of Table 7-20		

7.5.2.7 Reset Control Register (RSTCTRL)

This register contains a key that enables writes to the MSB of this register and the RSTCFG register. The key value is 0x5A69. A valid key will be stored as 0x000C, any other key value is invalid. When the RSTCTRL or the RSTCFG is written, the key is invalidated. Every write must be set up with a valid key. The Software Reset Control Register (RSTCTRL) is shown in Figure 7-14 and described in Table 7-21.

Figure 7-14 Reset Control Register (RSTCTRL)

31 17	16	15 0
Reserved	SWRST	KEY
R-0x0000	R/W-0x (1)	R/W-0x0003

Legend: R = Read only; -*n* = value after reset;

1 Writes are conditional based on valid key.

Table 7-21 Reset Control Register Field Descriptions

Bit	Field	Description
31-17	Reserved	Reserved
16	SWRST	Software reset 0 = Reset 1 = Not reset
15-0	KEY	Key used to enable writes to RSTCTRL and RSTCFG.
End of Table 7-21		

7.5.2.8 Reset Configuration Register (RSTCFG)

This register is used to configure the type of reset initiated by $\overline{\text{RESET}}$, the watchdog timer, and the PLL Controller's RSTCTRL Register; i.e., a hard reset or a soft reset. By default, these resets are hard resets. The Reset Configuration Register (RSTCFG) is shown in Figure 7-15 and described in Table 7-22.

Figure 7-15 Reset Configuration Register (RSTCFG)

31 16	15 14	13	12	11 4	3 0
Reserved	Reserved	PLLCTLRSTTYPE	$\overline{\text{RESET}}$TYPE	Reserved	WDTYPE[N (1)]
R-0x0000	R-00	R/W-0 (2)	R/W-0 2	R-0x0	R/W-0x00 2

Legend: R = Read only; R/W = Read/Write; -*n* = value after reset

1 Where N = 1, 2, 3,....N (Not all these output may be used on a specific device. For more information, see the device-specific data manual)
2 Writes are conditional based on valid key. For details, see Section 7.5.2.7 "Reset Control Register (RSTCTRL)".

Table 7-22 Reset Configuration Register Field Descriptions (Part 1 of 2)

Bit	Field	Description
31-14	Reserved	Reserved
13	PLLCTLRSTTYPE	PLL controller initiates a software-driven reset of type: 0 = Hard reset (default) 1 = Soft reset
12	$\overline{\text{RESET}}$TYPE	$\overline{\text{RESET}}$ initiates a reset of type: 0 = Hard reset (default) 1 = Soft reset

Table 7-22 Reset Configuration Register Field Descriptions (Part 2 of 2)

Bit	Field	Description
11-4	Reserved	Reserved
3 2 1 0	WDTYPE3 WDTYPE2 WDTYPE1 WDTYPE0	Watchdog timer [N] initiates a reset of type: 0 = Hard reset (default) 1 = Soft reset
End of Table 7-22		

7.5.2.9 Reset Isolation Register (RSISO)

This register is used to select the module clocks that must maintain their clocking without pausing through non power-on reset. Setting any of these bits effectively blocks reset to all PLLCTL Registers in order to maintain current values of PLL multiplier, divide ratios and other settings. Along with setting module specific bit in RSISO, the corresponding MDCTLx[12] bit also needs to be set in the PSC to reset-isolate a particular module. For more information on the MDCTLx Register see the *Power Sleep Controller (PSC) for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66. The Reset Isolation register (RSTCTRL) is shown in Figure 7-16 and described in Table 7-23.

Figure 7-16 Reset Isolation Register (RSISO)

31 16	15 10	9	8	7 4	3	2 0
Reserved	Reserved	SRIOISO	SRISO	Reserved	AIF2ISO	Reserved
R-0x0000	R-0x00	R/W-0	R/W-0	R-0x0	R/W-0	R-000

Legend: R = Read only; R/W = Read/Write; -*n* = value after reset

Table 7-23 Reset Isolation Register Field Descriptions

Bit	Field	Description
31-10	Reserved	Reserved.
9	SRIOISO	Isolate SRIO module control 0 = Not reset isolated 1 = Reset isolated
8	SRISO	Isolate SmartReflex control 0 = Not reset isolated 1 = Reset isolated
7-4	Reserved	Reserved
3	AIF2ISO	Isolate AIF2 module control 0 = Not reset isolated 1 = Reset isolated
2-0	Reserved	Reserved
End of Table 7-23		

7.5.3 Main PLL Control Registers

The Main PLL uses two chip-level registers (MAINPLLCTL0 and MAINPLLCTL1) along with the PLL Controller for its configuration. These MMRs (memory-mapped registers) exist inside the Bootcfg space. To write to these registers, software should go through an un-locking sequence using KICK0/KICK1 registers. For valid configurable values of the MAINPLLCTL registers, see Section 2.4.3 "PLL Settings" on page 35. See Section 3.3.4 "Kicker Mechanism (KICK0 and KICK1) Register" on page 73 for the address location of the KICK registers and their locking and unlocking sequences. These registers reset only on a $\overline{\text{POR}}$ reset. See Figure 7-17 and Table 7-24 for MAINPLLCTL0 details and Figure 7-18 and Table 7-25 for MAINPLLCTL1 details.

Figure 7-17 Main PLL Control Register (MAINPLLCTL0)

31 24	23 19	18 12	11 6	5 0
BWADJ[7:0]	Reserved	PLLM[12:6]	Reserved	PLLD
RW,+0000 0101	RW - 0000 0	RW,+0000000	RW, +000000	RW,+000000

Legend: RW = Read/Write; -*n* = value after reset

Table 7-24 Main PLL Control Register (MAINPLLCTL0) Field Descriptions

Bit	Field	Description
31-24	BWADJ[7:0]	BWADJ[11:8] and BWADJ[7:0] are located in MAINPLLCTL0 and MAINPLLCTL1 registers. BWADJ[11:0] should be programmed to a value equal to half of PLLM[12:0] value (round down if PLLM has an odd value) Example: If PLLM = 15, then BWADJ = 7
23-19	Reserved	Reserved
18-12	PLLM[12:6]	A 13-bit field that selects the values for the multiplication factor (see note below)
11-6	Reserved	Reserved
5-0	PLLD	A 6-bit field that selects the values for the reference divider
End of Table 7-24		

Figure 7-18 Main PLL Control Register (MAINPLLCTL1)

31 7	6	5 4	3 0
Reserved	ENSAT	Reserved	BWADJ[11:8]
RW - 00000000000000000000000000	RW - 0	RW- 00	RW- 0000

Legend: RW = Read/Write; -*n* = value after reset

Table 7-25 Main PLL Control Register (MAINPLLCTL1) Field Descriptions

Bit	Field	Description
31-7	Reserved	Reserved
6	ENSAT	Needs to be set to 1 for proper operation of the Main PLL
5-4	Reserved	Reserved
3-0	BWADJ[11:8]	BWADJ[11:8] and BWADJ[7:0] are located in MAINPLLCTL0 and MAINPLLCTL1 registers. BWADJ[11:0] should be programmed to a value equal to half of PLLM[12:0] value (round down if PLLM has an odd value) Example: If PLLM = 15, then BWADJ = 7
End of Table 7-25		

Note—The PLLM[5:0] bits of the multiplier are controlled by the PLLM register inside the PLL Controller and the PLLM[12:6] bits are controlled by the above chip-level register. MAINPLLCTL0 register PLLM[12:6] bits should be written just before writing to PLLM register PLLM[5:0] bits in the controller to have the complete 13 bit value latched when the GO operation is initiated in the PLL controller. See the *Phase Locked Loop (PLL) Controller for KeyStone Devices User Guide* in "Related Documentation from Texas Instruments" on page 66 for the recommended programming sequence. Output Divide ratio and Bypass enable/disable of the Main PLL is also controlled by the SECCTL register in the PLL Controller. See the "PLL Secondary Control Register (SECCTL)" on page 132 for more details.

7.5.4 Main PLL and PLL Controller Initialization Sequence

See the *Phase Locked Loop (PLL) Controller for KeyStone Devices User Guide* in "Related Documentation from Texas Instruments" on page 66 for details on the initialization sequence for Main PLL and PLL Controller.

7.5.5 Main PLL Controller/SRIO/HyperLink/PCIe Clock Input Electrical Data/Timing

Table 7-26 Main PLL Controller/SRIO/HyperLink/PCIe Clock Input Timing Requirements (1)
(see Figure 7-19 and Figure 7-20)

No.			Min	Max	Unit
		SYSCLK[P:N]			
1	tc(SYSCLKN)	Cycle time SYSCLKN cycle time	3.25 or 6.51 or 8.138 (2)		ns
1	tc(SYSCLKP)	Cycle time SYSCLKP cycle time	3.25 or 6.51 or 8.138		ns
3	tw(SYSCLKN)	Pulse width SYSCLKN high	0.45*tc	0.55*tc	ns
2	tw(SYSCLKN)	Pulse width SYSCLKN low	0.45*tc	0.55*tc	ns
2	tw(SYSCLKP)	Pulse width SYSCLKP high	0.45*tc	0.55*tc	ns
3	tw(SYSCLKP)	Pulse width SYSCLKP low	0.45*tc	0.55*tc	ns
4	tr(SYSCLKN_250 mv)	Transition time SYSCLKN rise time (250 mV)	50	350	ps
4	tf(SYSCLKN_250 mv)	Transition time SYSCLKN fall time (250 mV)	50	350	ps
4	tr(SYSCLKP_250 mv)	Transition time SYSCLKP rise time (250 mV)	50	350	ps
4	tf(SYSCLKP_250 mv)	Transition time SYSCLKP fall time (250 mV)	50	350	ps
5	tj(SYSCLKN)	Jitter, peak_to_peak _ periodic SYSCLKN		100 (4 (3))	ps
5	tj(SYSCLKP)	Jitter, peak_to_peak _ periodic SYSCLKP		100 (4)	ps
		ALTCORECLK[P:N]			
1	tc(ALTCORCLKN)	Cycle time ALTCORECLKN cycle time	3.2	25	ns
1	tc(ALTCORECLKP)	Cycle time ALTCORECLKP cycle time	3.2	25	ns
3	tw(ALTCORECLKN)	Pulse width ALTCORECLKN high	0.45*tc(ALTCORECLKN)	0.55*tc(ALTCORECLKN)	ns
2	tw(ALTCORECLKN)	Pulse width ALTCORECLKN low	0.45*tc(ALTCORECLKN)	0.55*tc(ALTCORECLKN)	ns
2	tw(ALTCORECLKP)	Pulse width ALTCORECLKP high	0.45*tc(ALTCORECLKP)	0.55*tc(ALTCORECLKP)	ns
3	tw(ALTCORECLKP)	Pulse width ALTCORECLKP low	0.45*tc(ALTCORECLKP)	0.55*tc(ALTCORECLKP)	ns
4	tr(ALTCORECLKN_250 mv)	Transition time ALTCORECLKN rise time (250 mV)	50	350	ps
4	tf(ALTCORECLKN_250 mv)	Transition time ALTCORECLKN fall time (250 mV)	50	350	ps
4	tr(ALTCORECLKP_250 mv)	Transition time ALTCORECLKP rise time (250 mV)	50	350	ps
4	tf(ALTCORECLKP_250 mv)	Transition time ALTCORECLKP fall time (250 mV)	50	350	ps
5	tj(ALTCORECLKN)	Jitter, peak_to_peak _ periodic ALTCORECLKN		100	ps
5	tj(ALTCORECLKP)	Jitter, peak_to_peak _ periodic ALTCORECLKP		100	ps

Table 7-26 Main PLL Controller/SRIO/HyperLink/PCIe Clock Input Timing Requirements (1)
(see Figure 7-19 and Figure 7-20)

No.			Min	Max	Unit
SRIOSGMIICLK[P:N]					
1	tc(SRIOSMGMIICLKN)	Cycle time SRIOSMGMIICLKN cycle time	3.2 or 4 or 6.4		ns
1	tc(SRIOSMGMIICLKP)	Cycle time SRIOSMGMIICLKP cycle time	3.2 or 4 or 6.4		ns
3	tw(SRIOSMGMIICLKN)	Pulse width SRIOSMGMIICLKN high	0.45*tc(SRIOSGMIICLKN)	0.55*tc(SRIOSGMIICLKN)	ns
2	tw(SRIOSMGMIICLKN)	Pulse width SRIOSMGMIICLKN low	0.45*tc(SRIOSGMIICLKN)	0.55*tc(SRIOSGMIICLKN)	ns
2	tw(SRIOSMGMIICLKP)	Pulse width SRIOSMGMIICLKP high	0.45*tc(SRIOSGMIICLKP)	0.55*tc(SRIOSGMIICLKP)	ns
3	tw(SRIOSMGMIICLKP)	Pulse width SRIOSMGMIICLKP low	0.45*tc(SRIOSGMIICLKP)	0.55*tc(SRIOSGMIICLKP)	ns
4	tr(SRIOSMGMIICLKN_250mv)	Transition time SRIOSMGMIICLKN rise time (250 mV)	50	350	ps
4	tf(SRIOSMGMIICLKN_250mv)	Transition time SRIOSMGMIICLKN fall time (250 mV)	50	350	ps
4	tr(SRIOSMGMIICLKP_250mv)	Transition time SRIOSMGMIICLKP rise time (250 mV)	50	350	ps
4	tf(SRIOSMGMIICLKP_250mv)	Transition time SRIOSMGMIICLKP fall time (250 mV)	50	350	ps
5	tj(SRIOSMGMIICLKN)	Jitter, RMS SRIOSMGMIICLKN		4	ps, RMS
5	tj(SRIOSMGMIICLKP)	Jitter, RMS SRIOSMGMIICLKP		4	ps, RMS
5	tj(SRIOSMGMIICLKN)	Jitter, RMS SRIOSMGMIICLKN (SRIO not used)		8	ps, RMS
5	tj(SRIOSMGMIICLKP)	Jitter, RMS SRIOSMGMIICLKP (SRIO not used)		8	ps, RMS
HyperLink CLK[P:N]					
1	tc(MCMCLKN)	Cycle time MCMCLKN cycle time	3.2 or 4 or 6.4		ns
1	tc(MCMCLKP)	Cycle time MCMCLKP cycle time	3.2 or 4 or 6.4		ns
3	tw(MCMCLKN)	Pulse width MCMCLKN high	0.45*tc(MCMCLKN)	0.55*tc(MCMCLKN)	ns
2	tw(MCMCLKN)	Pulse width MCMCLKN low	0.45*tc(MCMCLKN)	0.55*tc(MCMCLKN)	ns
2	tw(MCMCLKP)	Pulse width MCMCLKP high	0.45*tc(MCMCLKP)	0.55*tc(MCMCLKP)	ns
3	tw(MCMCLKP)	Pulse width MCMCLKP low	0.45*tc(MCMCLKP)	0.55*tc(MCMCLKP)	ns
4	tr(MCMCLKN_250mv)	Transition time MCMCLKN rise time (250 mV)	50	350	ps
4	tf(MCMCLKN_250mv)	Transition time MCMCLKN fall time (250 mV)	50	350	ps
4	tr(MCMCLKP_250mv)	Transition time MCMCLKP rise time (250 mV)	50	350	ps
4	tf(MCMCLKP_250mv)	Transition time MCMCLKP fall time (250 mV)	50	350	ps
5	tj(MCMCLKN)	Jitter, RMS MCMCLKN		4	ps, RMS
5	tj(MCMCLKP)	Jitter, RMS MCMCLKP		4	ps, RMS
PCIECLK[P:N]					
1	tc(PCIECLKN)	Cycle time PCIECLKN cycle time	3.2 or 4 or 6.4 or 10		ns
1	tc(PCIECLKP)	Cycle time PCIECLKP cycle time	3.2 or 4 or 6.4 or 10		ns
3	tw(PCIECLKN)	Pulse width PCIECLKN high	0.45*tc(PCIECLKN)	0.55*tc(PCIECLKN)	ns
2	tw(PCIECLKN)	Pulse width PCIECLKN low	0.45*tc(PCIECLKN)	0.55*tc(PCIECLKN)	ns
2	tw(PCIECLKP)	Pulse width PCIECLKP high	0.45*tc(PCIECLKP)	0.55*tc(PCIECLKP)	ns
3	tw(PCIECLKP)	Pulse width PCIECLKP low	0.45*tc(PCIECLKP)	0.55*tc(PCIECLKP)	ns
4	tr(PCIECLKN_250mv)	Transition time PCIECLKN rise time (250 mV)	50	350	ps
4	tf(PCIECLKN_250mv)	Transition time PCIECLKN fall time (250 mV)	50	350	ps
4	tr(PCIECLKP_250mv)	Transition time PCIECLKP rise time (250 mV)	50	350	ps
4	tf(PCIECLKP_250mv)	Transition time PCIECLKP fall time (250 mV)	50	350	ps

Table 7-26 Main PLL Controller/SRIO/HyperLink/PCIe Clock Input Timing Requirements (1)
(see Figure 7-19 and Figure 7-20)

No.			Min	Max	Unit
5	tj(PCIECLKN)	Jitter, RMS PCIECLKN		4	ps, RMS
5	tj(PCIECLKP)	Jitter, RMS PCIECLKP		4	ps, RMS
End of Table 7-26					

1 See the Hardware Design Guide for KeyStone Devices in 2.9 "Related Documentation from Texas Instruments" on page 66 for detailed recommendations.
2 If AIF2 is being used then SYSCLK(N|P) can be programmed only to fixed values, if AIF2 is not being used then any value in the range between the min and max values can be used.
3 If AIF2 is used then the Max allowed jitter on SYSCLK(N|P) is 4ps RMS

Figure 7-19 Main PLL Controller/SRIO/HyperLink/PCIe Clock Input Timing

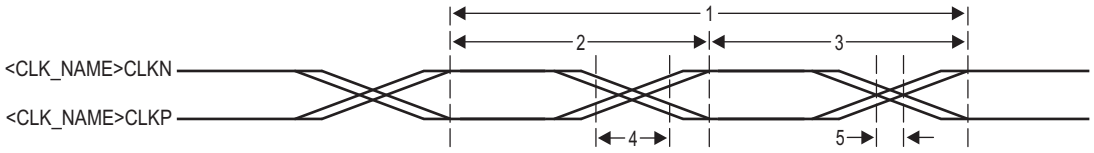
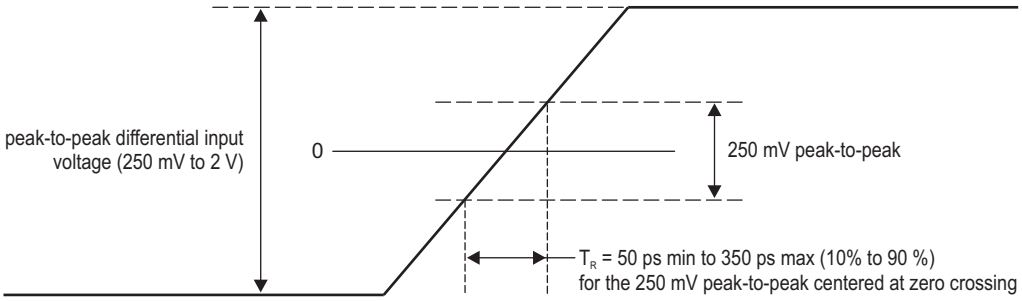


Figure 7-20 Main PLL Transition Time

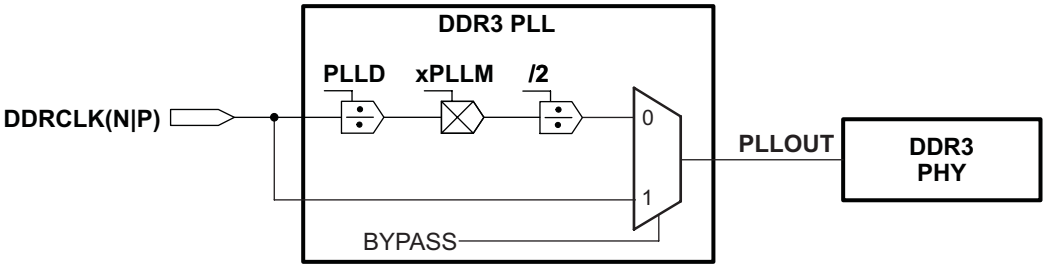


7.6 DDR3 PLL

The DDR3 PLL generates interface clocks for the DDR3 memory controller. When coming out of power-on reset, DDR3 PLL is programmed to a valid frequency during the boot config before being enabled and used.

DDR3 PLL power is supplied via the DDR3 PLL power-supply pin (AVDDA2). An external EMI filter circuit must be added to all PLL supplies. See the *Hardware Design Guide for KeyStone Devices* in 2.9 "Related Documentation from Texas Instruments" on page 66 for detailed recommendations. For the best performance, TI recommends that all the PLL external components be on a single side of the board without jumpers, switches, or components other than those shown. For reduced PLL jitter, maximize the spacing between switching signal traces and the PLL external components (C1, C2, and the EMI Filter).

Figure 7-21 DDR3 PLL Block Diagram



7.6.1 DDR3 PLL Control Registers

The DDR3 PLL, which is used to drive the DDR PHY for the EMIF, does not use a PLL controller. DDR3 PLL can be controlled using the DDR3PLLCTL0 and DDR3PLLCTL1 registers located in the Bootcfg module. This MMR (memory-mapped register) exists inside the Bootcfg space. To write to these registers, software should go through an un-locking sequence using KICK0/KICK1 registers. For suggested configurable values see 2.4.3 "PLL Settings" on page 35. See 3.3.4 "Kicker Mechanism (KICK0 and KICK1) Register" on page 73 for the address location of the registers and locking and unlocking sequences for accessing the registers. These registers are reset on $\overline{\text{POR}}$ only.

Figure 7-22 DDR3 PLL Control Register (DDR3PLLCTL0) [1]

31 24	23	22 19	18 6	5 0
BWADJ[7:0]	BYPASS	Reserved	PLLM	PLLD
RW,+0000 1001	RW,+0	RW,+0001	RW,+0000000010011	RW,+000000

Legend: RW = Read/Write; -*n* = value after reset

1 This register is Reset on POR only

Table 7-27 DDR3 PLL Control Register 0 Field Descriptions

Bit	Field	Description
31-24	BWADJ[7:0]	BWADJ[11:8] and BWADJ[7:0] are located in DDR3PLLCTL0 and DDR3PLLCTL1 registers. BWADJ[11:0] should be programmed to a value equal to half of PLLM[12:0] value (round down if PLLM has an odd value) Example: If PLLM = 15, then BWADJ = 7
23	BYPASS	Enable bypass mode 0 = Bypass disabled 1 = Bypass enabled
22-19	Reserved	Reserved
18-6	PLLM	A 13-bit field that selects the values for the multiplication factor
5-0	PLLD	A 6-bit field that selects the values for the reference divider
End of Table 7-27		

Figure 7-23 DDR3 PLL Control Register 1 (DDR3PLLCTL1)

31 14	13	12 7	6	5 4	3 0
Reserved	PLLRST	Reserved	ENSAT	Reserved	BWADJ[11:8]
RW-00000000000000000	RW-0	RW-000000	RW-0	R-0	RW-0000

Legend: RW = Read/Write; -*n* = value after reset

Table 7-28 DDR3 PLL Control Register 1 Field Descriptions (DDR3PLLCTL1)

Bit	Field	Description
31-14	Reserved	Reserved
13	PLLRST	PLL reset bit. 0 = PLL reset is released. 1 = PLL reset is asserted.
12-7	Reserved	Reserved
6	ENSAT	Needs to be set to 1 for proper operation of PLL
5-4	Reserved	Reserved
3-0	BWADJ[11:8]	BWADJ[11:8] and BWADJ[7:0] are located in DDR3PLLCTL0 and DDR3PLLCTL1 registers. BWADJ[11:0] should be programmed to a value equal to half of PLLM[12:0] value (round down if PLLM has an odd value) Example: If PLLM = 15, then BWADJ = 7
End of Table 7-28		

7.6.2 DDR3 PLL Device-Specific Information

As shown in Figure 7-21, the output of DDR3 PLL (PLLOUT) is divided by 2 and directly fed to the DDR3 memory controller. The DDR3 PLL is affected by power-on reset. During power-on resets, the internal clocks of the DDR3 PLL are affected as described in Section 7.4 "Reset Controller" on page 122. DDR3 PLL is unlocked only during the power-up sequence and is locked by the time the $\overline{\text{RESETSTAT}}$ pin goes high. It does not lose lock during any of the other resets.

7.6.3 DDR3 PLL Initialization Sequence

The Main PLL and PLL Controller must always be initialized prior to initializing the DDR3 PLL. The sequence shown below must be followed to initialize the DDR3 PLL.

1. In DDR3PLLCTL1, write ENSAT = 1 (for optimal PLL operation)
2. In DDR3PLLCTL0, write BYPASS = 1 (set the PLL in Bypass)
3. In DDR3PLLCTL1, write PLLRST = 1 (PLL is reset)
4. Program PLLM and PLLD in DDR3PLLCTL0 register

5. Program BWADJ[7:0] in DDR3PLLCTL0 and BWADJ[11:8] in DDR3PLLCTL1 register. BWADJ value must be set to ((PLLM + 1) >> 1) - 1)
6. Wait for at least 5 µs based on the reference clock (PLL reset time)
7. In DDR3PLLCTL1, write PLLRST = 0 (PLL reset is released)
8. Wait for at least 500 *REFCLK cycles * (PLLD + 1) (PLL lock time)
9. In DDR3PLLCTL0, write BYPASS = 0 (switch to PLL mode)

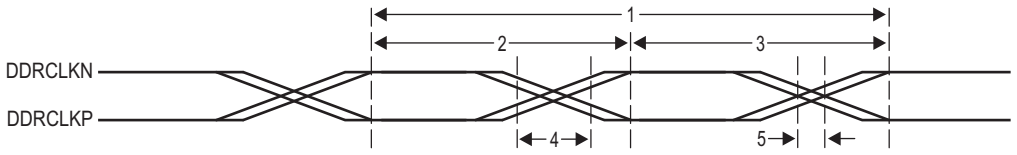
CAUTION—Software must always perform Read-modify-write to any register in the PLL. This is to ensure that only the relevant bits in the register are modified and the rest of the bits including the reserved bits are not affected.

7.6.4 DDR3 PLL Input Clock Electrical Data/Timing

Table 7-29 DDR3 PLL DDRCLK(N|P) Timing Requirements
(see Figure 7-24 and Figure 7-20)

No.			Min	Max	Unit
DDRCLK[P:N]					
1	tc(DDRCLKN)	Cycle time _ DDRCLKN cycle time	3.2	25	ns
1	tc(DDRCLKP)	Cycle time _ DDRCLKP cycle time	3.2	25	ns
3	tw(DDRCLKN)	Pulse width _ DDRCLKN high	0.45*tc(DDRCLKN)	0.55*tc(DDRCLKN)	ns
2	tw(DDRCLKN)	Pulse width _ DDRCLKN low	0.45*tc(DDRCLKN)	0.55*tc(DDRCLKN)	ns
2	tw(DDRCLKP)	Pulse width _ DDRCLKP high	0.45*tc(DDRCLKP)	0.55*tc(DDRCLKP)	ns
3	tw(DDRCLKP)	Pulse width _ DDRCLKP low	0.45*tc(DDRCLKP)	0.55*tc(DDRCLKP)	ns
4	tr(DDRCLKN_250 mv)	Transition time _ DDRCLKN rise time (250 mV)	50	350	ps
4	tf(DDRCLKN_250 mv)	Transition time _ DDRCLKN fall time (250 mV)	50	350	ps
4	tr(DDRCLKP_250 mv)	Transition time _ DDRCLKP rise time (250 mV)	50	350	ps
4	tf(DDRCLKP_250 mv)	Transition time _ DDRCLKP fall time (250 mV)	50	350	ps
5	tj(DDRCLKN)	Jitter, peak_to_peak _ periodic DDRCLKN		0.025*tc(DDRCLKN)	ps
5	tj(DDRCLKP)	Jitter, peak_to_peak _ periodic DDRCLKP		0.025*tc(DDRCLKP)	ps
End of Table 7-29					

Figure 7-24 DDR3 PLL DDRCLK Timing

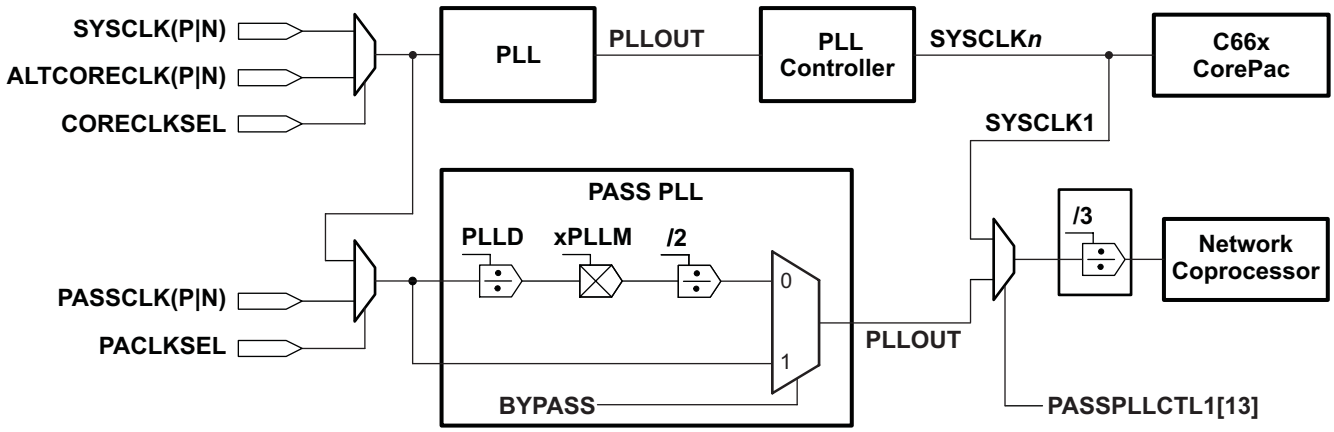


7.7 PASS PLL

The PASS PLL generates interface clocks for the Network Coprocessor. Using the PACLKSEL pin the user can select the input source of PASS PLL as either the output of Main PLL mux or the PASSCLK clock reference sources. When coming out of power-on reset, PASS PLL comes out in a bypass mode and needs to be programmed to a valid frequency before being enabled and used.

PASS PLL power is supplied via the PASS PLL power-supply pin (AVDDA3). An external EMI filter circuit must be added to all PLL supplies. Please see the *Hardware Design Guide for KeyStone Devices* in 2.9 "Related Documentation from Texas Instruments" on page 66 for detailed recommendations. For the best performance, TI recommends that all the PLL external components be on a single side of the board without jumpers, switches, or components other than those shown. For reduced PLL jitter, maximize the spacing between switching signal traces and the PLL external components (C1, C2, and the EMI Filter).

Figure 7-25 PASS PLL Block Diagram



7.7.1 PASS PLL Control Registers

The PASS PLL, which is used to drive the Network Coprocessor, does not use a PLL controller. PASS PLL can be controlled using the PAPLLCTL0 and PAPLLCTL1 registers located in Bootcfg module. These MMRs (memory-mapped registers) exist inside the Bootcfg space. To write to these registers, software should go through an un-locking sequence using KICK0/KICK1 registers. For suggested configurable values see 2.4.3 "PLL Settings" on page 35. See 3.3.4 "Kicker Mechanism (KICK0 and KICK1) Register" on page 73 for the address location of the registers and locking and unlocking sequences for accessing these registers. These registers are reset on $\overline{\text{POR}}$ only.

Figure 7-26 PASS PLL Control Register (PASSPLLCTL0) [1]

31 — 24	23	22 — 19	18 — 6	5 — 0
BWADJ[7:0]	BYPASS	Reserved	PLLM	PLLD
RW,+0000 1001	RW,+0	RW,+0001	RW,+0000000010011	RW,+000000

Legend: RW = Read/Write; *-n* = value after reset

1 This register is Reset on POR only.

Table 7-30 PASS PLL Control Register 0 Field Descriptions (PASSPLLCTL0)

Bit	Field	Description
31-24	BWADJ[7:0]	BWADJ[11:8] and BWADJ[7:0] are located in PASSPLLCTL0 and PASSPLLCTL1 registers. BWADJ[11:0] should be programmed to a value equal to half of PLLM[12:0] value (round down if PLLM has an odd value) Example: If PLLM = 15, then BWADJ = 7
23	BYPASS	Enable bypass mode 0 = Bypass disabled 1 = Bypass enabled
22-19	Reserved	Reserved
18-6	PLLM	A 13-bit field that selects the values for the multiplication factor (see note below)
5-0	PLLD	A 6-bit field that selects the values for the reference divider
End of Table 7-30		

Figure 7-27 PASS PLL Control Register 1 (PASSPLLCTL1)

31 15	14	13	12 7	6	5 4	3 0
Reserved	PLLRST	PLLSELECT	Reserved	ENSAT	Reserved	BWADJ[11:8]
RW-00000000000000000	RW-0	0	RW-0000000	RW-0	R-0	RW-0000

Legend: RW = Read/Write; -*n* = value after reset

Table 7-31 PASS PLL Control Register 1 Field Descriptions (PASSPLLCTL1)

Bit	Field	Description
31-15	Reserved	Reserved
14	PLLRST	PLL reset bit 0 = PLL reset is released. 1 = PLL reset is asserted.
13	PLLSELECT	PASS PLL Select Bit 0 - Reserved 1 - PASS PLL output clock is used as the input to PASS
12-7	Reserved	Reserved
6	ENSAT	Needs to be set to 1 for proper operation of PLL
5-4	Reserved	Reserved
3-0	BWADJ[11:8]	BWADJ[11:8] and BWADJ[7:0] are located in PASSPLLCTL0 and PASSPLLCTL1 registers. BWADJ[11:0] should be programmed to a value equal to half of PLLM[12:0] value (round down if PLLM has an odd value) Example: If PLLM = 15, then BWADJ = 7
End of Table 7-31		

7.7.2 PASS PLL Device-Specific Information

As shown in Figure 7-25, the output of PASS PLL (PLLOUT) is divided by 3 and directly fed to the Network Coprocessor. The PASS PLL is affected by power-on reset. During power-on resets, the internal clocks of the PASS PLL are affected as described in Section 7.4 "Reset Controller" on page 122. The PASS PLL is unlocked only during the power-up sequence and is locked by the time the $\overline{\text{RESETSTAT}}$ pin goes high. It does not lose lock during any of the other resets.

7.7.3 PASS PLL Initialization Sequence

The Main PLL and PLL Controller must always be initialized prior to initializing the PASS PLL. The sequence shown below must be followed to initialize the PASS PLL.

1. In PASSPLLCTL1, write ENSAT = 1 (for optimal PLL operation)
2. In PASSPLLCTL0, write BYPASS = 1 (set the PLL in Bypass)

3. In PASSPLLCTL1, write PLLRST = 1 (PLL is reset)
4. Program PLLM and PLLD in PASSPLLCTL0 register
5. Program BWADJ[7:0] in PASSPLLCTL0 and BWADJ[11:8] in PASSPLLCTL1 register. BWADJ value must be set to ((PLLM + 1) >> 1) - 1)
6. Wait for at least 5 µs based on the reference clock (PLL reset time)
7. In PASSPLLCTL1, write PLLSELECT = 1 (for selecting the output of PASS PLL as the input to PASS)
8. In PASSPLLCTL1, write PLLRST = 0 (PLL reset is released)
9. Wait for at least 500 * REFCLK cycles * (PLLD + 1) (PLL lock time)
10. In PASSPLLCTL0, write BYPASS = 0 (switch to PLL mode)

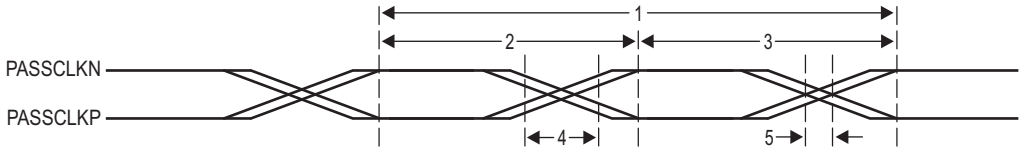
CAUTION—Software must always perform Read-modify-write to any register in the PLL. This is to ensure that only the relevant bits in the register are modified and the rest of the bits including the reserved bits are not affected.

7.7.4 PASS PLL Input Clock Electrical Data/Timing

Table 7-32 PASS PLL Timing Requirements
(See Figure 7-28 and Figure 7-20)

No.			Min	Max	Unit
PASSCLK[P:N]					
1	tc(PASSCLKN)	Cycle time _ PASSCLKN cycle time	3.2	6.4	ns
1	tc(PASSCLKP)	Cycle time _ PASSCLKP cycle time	3.2	6.4	ns
3	tw(PASSCLKN)	Pulse width _ PASSCLKN high	0.45*tc(PASSCLKN)	0.55*tc(PASSCLKN)	ns
2	tw(PASSCLKN)	Pulse width _ PASSCLKN low	0.45*tc(PASSCLKN)	0.55*tc(PASSCLKN)	ns
2	tw(PASSCLKP)	Pulse width _ PASSCLKP high	0.45*tc(PASSCLKP)	0.55*tc(PASSCLKP)	ns
3	tw(PASSCLKP)	Pulse width _ PASSCLKP low	0.45*tc(PASSCLKP)	0.55*tc(PASSCLKP)	ns
4	tr(PASSCLKN_250mv)	Transition time _ PASSCLKN rise time (250 mV)	50	350	ps
4	tf(PASSCLKN_250mv)	Transition time _ PASSCLKN fall time (250 mV)	50	350	ps
4	tr(PASSCLKP_250mv)	Transition time _ PASSCLKP rise time (250 mV)	50	350	ps
4	tf(PASSCLKP_250mv)	Transition time _ PASSCLKP fall time (250 mV)	50	350	ps
5	tj(PASSCLKN)	Jitter, peak_to_peak _ periodic PASSCLKN		100	ps, pk-pk
5	tj(PASSCLKP)	Jitter, peak_to_peak _ periodic PASSCLKP		100	ps, pk-pk

Figure 7-28 PASS PLL Timing



7.8 Enhanced Direct Memory Access (EDMA3) Controller

The primary purpose of the EDMA3 is to service user-programmed data transfers between two memory-mapped slave endpoints on the device. The EDMA3 services software-driven paging transfers (e.g., data movement between external memory and internal memory), performs sorting or subframe extraction of various data structures, services event driven peripherals, and offloads data transfers from the device CPU.

There are 3 EDMA channel controllers on the device: EDMA3CC0, EDMA3CC1, and EDMA3CC2.

- EDMA3CC0 has two transfer controllers: EDMA3TC1 andEDMA3TC2.
- EDMA3CC1 has four transfer controllers: EDMA3TC0, EDMA3TC1, EDMA3TC2, and EDMA3TC3.
- EDMA3CC2 has four transfer controllers: EDMA3TC0, EDMA3TC1, EDMA3TC2, and EDMA3TC3.

In the context of this document, EDMA3TCx is associated with EDMA3CCy, and is referred to as EDMA3CCy TCx. Each of the transfer controllers has a direct connection to the switch fabric. Section 4.2 "Switch Fabric Connections Matrix" lists the peripherals that can be accessed by the transfer controllers.

EDMA3CC0 is optimized to be used for transfers to/from/within the MSMC and DDR-3 subsytems. The others are used for the remaining traffic.

Each EDMA3 channel controller includes the following features:

- Fully orthogonal transfer description
  - 3 transfer dimensions:
    - Array (multiple bytes)
    - Frame (multiple arrays)
    - Block (multiple frames)
  - Single event can trigger transfer of array, frame, or entire block
  - Independent indexes on source and destination
- Flexible transfer definition:
  - Increment or FIFO transfer addressing modes
  - Linking mechanism allows for ping-pong buffering, circular buffering, and repetitive/continuous transfers, all with no CPU intervention
  - Chaining allows multiple transfers to execute with one event
- 128 PaRAM entries for EDMA3CC0, 512 each for EDMA3CC1 and EDMA3CC2
  - Used to define transfer context for channels
  - Each PaRAM entry can be used as a DMA entry, QDMA entry, or link entry
- 16 DMA channels for EDMA3CC0, 64 each for EDMA3CC1 and EDMA3CC2
  - Manually triggered (CPU writes to channel controller register), external event triggered, and chain triggered (completion of one transfer triggers another)
- 8 Quick DMA (QDMA) channels per EDMA3CCx
  - Used for software-driven transfers
  - Triggered upon writing to a single PaRAM set entry
- Two transfer controllers and two event queues with programmable system-level priority for EDMA3CC0, four transfer controllers and four event queues with programmable system-level priority for each of EDMA3CC1 and EDMA3CC2
- Interrupt generation for transfer completion and error conditions

- Debug visibility
  - Queue watermarking/threshold allows detection of maximum usage of event queues
  - Error and status recording to facilitate debug

7.8.1 EDMA3 Device-Specific Information

The EDMA supports two addressing modes: constant addressing and increment addressing mode. Constant addressing mode is applicable to a very limited set of use cases; for most applications increment mode can be used. On the C6670 SoC, the EDMA can use constant addressing mode only with the enhanced Viterbi decoder coprocessor (VCP) and the enhanced turbo decoder coprocessor (TCP). Constant addressing mode is not supported by any other peripheral or internal memory in the DSP. Note that increment mode is supported by all peripherals, including VCP and TCP. For more information on these two addressing modes, see the *Enhanced Direct Memory Access 3 (EDMA3) for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

For the range of memory addresses that include EDMA3 channel controller (EDMA3CC) control registers and EDMA3 transfer controller (EDMA3TC) control register see Section 2.2 "Memory Map Summary" on page 21. For memory offsets and other details on EDMA3CC and EDMA3TC Control Registers entries, see the *Enhanced Direct Memory Access 3 (EDMA3) for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

7.8.2 EDMA3 Channel Controller Configuration

Table 7-33 provides the configuration for each of the EDMA3 channel controllers present on the device.

Table 7-33 EDMA3 Channel Controller Configuration

Description	EDMA3 CC0	EDMA3 CC1	EDMA3 CC2
Number of DMA channels in channel controller	16	64	64
Number of QDMA channels	8	8	8
Number of interrupt channels	16	64	64
Number of PaRAM set entries	128	512	512
Number of event queues	2	4	4
Number of transfer controllers	2	4	4
Memory protection existence	Yes	Yes	Yes
Number of memory protection and shadow regions	8	8	8
End of Table 7-33			

7.8.3 EDMA3 Transfer Controller Configuration

Each transfer controller on the device is designed differently based on considerations like performance requirements, system topology (like main TeraNet bus width, external memory bus width), etc. The parameters that determine the transfer controller configurations are:

- **FIFOSIZE:** Determines the size in bytes for the data FIFO that is the temporary buffer for the in-flight data. The data FIFO is where the read return data read by the TC read controller from the source endpoint is stored and subsequently written out to the destination endpoint by the TC write controller.
- **BUSWIDTH:** The width of the read and write data buses in bytes, for the TC read and write controller, respectively. This is typically equal to the bus width of the main TeraNet interface.
- **Default Burst Size (DBS):** The DBS is the maximum number of bytes per read/write command issued by a transfer controller.
- **DSTREGDEPTH:** This determines the number of destination FIFO register set. The number of destination FIFO register set for a transfer controller determines the maximum number of outstanding transfer requests.

All four parameters listed above are fixed by the design of the device.

Table 7-34 shows the configuration of each of the EDMA3 transfer controllers present on the device.

Table 7-34 EDMA3 Transfer Controller Configuration

Parameter	EDMA3CC0		EDMA3CC1				EDMA3CC2			
	TC0	TC1	TC0	TC1	TC2	TC3	TC0	TC1	TC2	TC3
FIFOSIZE	1024 bytes	1024 bytes	1024 bytes	512 bytes	1024 bytes	512 bytes	1024 bytes	512 bytes	512 bytes	1024 bytes
BUSWIDTH	32 bytes	32 bytes	16 bytes	16 bytes	16 bytes	16 bytes	16 bytes	16 bytes	16 bytes	16 bytes
DSTREGDEPTH	4 entries	4 entries	4 entries	4 entries	4 entries	4 entries	4 entries	4 entries	4 entries	4 entries
DBS	128 bytes	128 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes
End of Table 7-34										

7.8.4 EDMA3 Channel Synchronization Events

The EDMA3 supports up to 16 DMA channels for CC0, 64 each for CC1 and CC2 that can be used to service system peripherals and to move data between system memories. DMA channels can be triggered by synchronization events generated by system peripherals. The following tables lists the source of the synchronization event associated with each of the EDMA CC DMA channels. On the C6670, the association of each synchronization event and DMA channel is fixed and cannot be reprogrammed.

For more detailed information on the EDMA3 module and how EDMA3 events are enabled, captured, processed, prioritized, linked, chained, and cleared, etc., see the *Enhanced Direct Memory Access 3 (EDMA3) for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

Table 7-35 EDMA3CC0 Events for C6670

Event Number	Event	Event Description
0	TCP3D_C_REVT0	TCP3d_C Receive event0
1	TCP3D_C_REVT1	TCP3d_C Receive event1
2	FFTC_C_ERROR0	FFTC_C Error event and FFTC_C debug event
3	FFTC_C_ERROR1	FFTC_C Error event and FFTC_C debug event
4	FFTC_C_ERROR2	FFTC_C Error event and FFTC_C debug event
5	FFTC_C_ERROR3	FFTC_C Error event and FFTC_C debug event
6	CIC2_OUT40	Interrupt Controller output
7	CIC2_OUT41	Interrupt Controller output
8	CIC2_OUT0	Interrupt Controller output
9	CIC2_OUT1	Interrupt Controller output
10	CIC2_OUT2	Interrupt Controller output
11	CIC2_OUT3	Interrupt Controller output
12	CIC2_OUT4	Interrupt Controller output
13	CIC2_OUT5	Interrupt Controller output
14	CIC2_OUT6	Interrupt Controller output
15	CIC2_OUT7	Interrupt Controller output
End of Table 7-35		

Table 7-36 EDMA3CC1 Events for C6670 (Part 1 of 2)

Event Number	Event	Event Description
0	SPIINT0	SPI interrupt
1	SPIINT1	SPI interrupt
2	SPIXEVT	Transmit event
3	SPIREVT	Receive event
4	I2CREVT	I^2C receive event
5	I2CXEVT	I^2C transmit event
6	GPINT0	GPIO interrupt
7	GPINT1	GPIO interrupt
8	GPINT2	GPIO interrupt
9	GPINT3	GPIO interrupt
10	AIF_SEVT0	AIF radio timing sync event 0
11	AIF_SEVT1	AIF radio timing sync event 1
12	AIF_SEVT2	AIF radio timing sync event 2
13	AIF_SEVT3	AIF radio timing sync event 3
14	AIF_SEVT4	AIF radio timing sync event 4
15	AIF_SEVT5	AIF radio timing sync event 5
16	AIF_SEVT6	AIF radio timing sync event 6
17	AIF_SEVT7	AIF radio timing sync event 7
18	SEMINT0	Semaphore interrupt
19	SEMINT1	Semaphore interrupt
20	SEMINT2	Semaphore interrupt
21	SEMINT3	Semaphore interrupt
22	TINT4L	Timer interrupt low
23	TINT4H	Timer interrupt high
24	TINT5L	Timer interrupt low
25	TINT5H	Timer interrupt high
26	TINT6L	Timer interrupt low
27	TINT6H	Timer interrupt high
28	TINT7L	Timer interrupt low
29	TINT7H	Timer interrupt high
30	RAC_AINT0	RAC_A_ interrupt 0
31	RAC_AINT1	RAC_A_ interrupt 1
32	RAC_AINT2	RAC_A_interrupt 2
33	RAC_AINT3	RAC_A_interrupt 3
34	RAC_ADEVENT0	RAC_A_debug event
35	RAC_ADEVENT1	RAC_A_debug event
36	TAC_INTD	TAC error interrupt
37	TACDEVENT0	TAC debug event
38	TACDEVENT1	TAC debug event
39	RAC_BINT0	RAC_B_ interrupt 0
40	RAC_BINT1	RAC_B_ interrupt 1
41	RAC_BINT2	RAC_B_interrupt 2
42	RAC_BINT3	RAC_B_interrupt 3
43	RAC_BDEVENT0	RAC_B_debug Event

Table 7-36 EDMA3CC1 Events for C6670 (Part 2 of 2)

Event Number	Event	Event Description
44	RAC_BDEVENT1	RAC_B_debug Event
45	CIC1_OUT2	Interrupt Controller output
46	CIC1_OUT3	Interrupt Controller output
47	CIC1_OUT4	Interrupt Controller output
48	CIC1_OUT5	Interrupt Controller output
49	CIC1_OUT6	Interrupt Controller output
50	CIC1_OUT7	Interrupt Controller output
51	CIC1_OUT8	Interrupt Controller output
52	CIC1_OUT9	Interrupt Controller output
53	CIC1_OUT10	Interrupt Controller output
54	CIC1_OUT11	Interrupt Controller output
55	CIC1_OUT12	Interrupt Controller output
56	CIC1_OUT13	Interrupt Controller output
57	CIC1_OUT14	Interrupt Controller output
58	CIC1_OUT15	Interrupt Controller output
59	CIC1_OUT16	Interrupt Controller output
60	CIC1_OUT17	Interrupt Controller output
61	CIC1_OUT18	Interrupt Controller output
62	CIC1_OUT19	Interrupt Controller output
63	CIC1_OUT20	Interrupt Controller output
End of Table 7-36		

Table 7-37 EDMA3CC2 Events for C6670 (Part 1 of 3)

Event Number	Event	Event Description
0	TCP3D_AREVT0	TCP3D_A receive event0
1	TCP3D_AREVT1	TCP3D_A receive event1
2	TCP3EREVT	TCP3e read event
3	TCP3EWEVT	TCP3e write event
4	URXEVT	UART receive event
5	UTXEVT	UART transmit event
6	GPINT0	GPIO interrupt
7	GPINT1	GPIO interrupt
8	GPINT2	GPIO interrupt
9	GPINT3	GPIO interrupt
10	VCPAREVT	Receive event
11	VCPAXEVT	Transmit event
12	VCPBREVT	Receive event
13	VCPBXEVT	Transmit event
14	VCPCREVT	Receive event
15	VCPCXEVT	Transmit event
16	VCPDREVT	Receive event
17	VCPDXEVT	Transmit event
18	SEMINT0	Semaphore interrupt
19	SEMINT1	Semaphore interrupt

Table 7-37 EDMA3CC2 Events for C6670 (Part 2 of 3)

Event Number	Event	Event Description
20	SEMINT2	Semaphore interrupt
21	SEMINT3	Semaphore interrupt
22	TINT4L	Timer interrupt low
23	TINT4H	Timer interrupt high
24	TINT5L	Timer interrupt low
25	TINT5H	Timer interrupt high
26	TINT6L	Timer interrupt low
27	TINT6H	Timer interrupt high
28	TINT7L	Timer interrupt low
29	TINT7H	Timer interrupt high
30	SPIXEVT	SPI transmit event
31	SPIREVT	SPI receive event
32	I2CREVT	I^2C receive event
33	I2CXEVT	I^2C transmit event
34	TCP3D_BREVT0	TCP3D_B receive event0
35	TCP3D_BREVT1	TCP3D_B receive event1
36	CIC1_OUT23	Interrupt Controller output
37	CIC1_OUT24	Interrupt Controller output
38	CIC1_OUT25	Interrupt Controller output
39	CIC1_OUT26	Interrupt Controller output
40	CIC1_OUT27	Interrupt Controller output
41	CIC1_OUT28	Interrupt Controller output
42	CIC1_OUT29	Interrupt Controller output
43	CIC1_OUT30	Interrupt Controller output
44	CIC1_OUT31	Interrupt Controller output
45	CIC1_OUT32	Interrupt Controller output
46	CIC1_OUT33	Interrupt Controller output
47	CIC1_OUT34	Interrupt Controller output
48	CIC1_OUT35	Interrupt Controller output
49	CIC1_OUT36	Interrupt Controller output
50	CIC1_OUT37	Interrupt Controller output
51	CIC1_OUT38	Interrupt Controller output
52	CIC1_OUT39	Interrupt Controller output
53	CIC1_OUT40	Interrupt Controller output
54	CIC1_OUT41	Interrupt Controller output
55	CIC1_OUT42	Interrupt Controller output
56	CIC1_OUT43	Interrupt Controller output
57	CIC1_OUT44	Interrupt Controller output
58	TCP3D_C_REVT0	TCP3d_C Receive event0
59	TCP3D_C_REVT1	TCP3d_C Receive event1
60	FFTC_C_ERROR0	FFTC_C Error event and FFTC_C debug event
61	FFTC_C_ERROR1	FFTC_C Error event and FFTC_C debug event

Table 7-37 EDMA3CC2 Events for C6670 (Part 3 of 3)

Event Number	Event	Event Description
62	FFTC_C_ERROR2	FFTC_C Error event and FFTC_C debug event
63	FFTC_C_ERROR3	FFTC_C Error event and FFTC_C debug event
End of Table 7-37		

7.9 Interrupts

7.9.1 Interrupt Sources and Interrupt Controller

The CPU interrupts on the C6670 device are configured through the C66x CorePac Interrupt Controller. The Interrupt Controller allows for up to 128 system events to be programmed to any of the twelve CPU interrupt inputs (CPUINT4 - CPUINT15), the CPU exception input (EXCEP), or the advanced emulation logic. The 128 system events consist of 25 internally-generated events (within the CorePac) and 103 chip-level events.

Additional system events are routed to each of the C66x CorePacs to provide chip-level events that are not required as CPU interrupts/exceptions to be routed to the Interrupt Controller as emulation events. In addition, error-class events or infrequently used events are also routed through the system event router to offload the C66x CorePac interrupt selector. This is accomplished through Chip Interrupt Controller (CIC) blocks, CIC[2:0] for C6670 device. This is clocked using CPU/6.

The event controllers consist of simple combination logic to provide additional events to each C66x CorePac, plus the EDMA3CC. CIC0 provides 26 additional events (18 that are CorePac-specific plus 8 broadcast) to each of the C66x CorePacs, CIC1 provides 19 and 21 additional events to CC1 and CC2, respectively, and CIC2 provides 10 and 32 additional events to CC0 and HyperLink, respectively.

The events that are routed to the C66x CorePacs for AET purposes, from those EDMA3CC and FSYNC events that are not otherwise provided to each C66x CorePac. For more details on the CIC features, please see the *Chip Interrupt Controller (CIC) for KeyStone Devices User Guid*e in 2.9 "Related Documentation from Texas Instruments" on page 66.

Note—Modules such as FFTC, TCP3d, TCP3e, TAC, AIF, CP_MPU, BOOT_CFG, and Tracer have level interrupts and EOI handshaking interface. The EOI value is 0 for TCP3d, TCP3e, TAC, AIF, CP_MPU, BOOT_CFG, and Tracer. For FFTC, the EOI values are 0 for FFTC_x_INTD0, 1 for FFTC_x_INTD01, 2 for FFTC_x_INTD2, and 3 for FFTC_x_INTD3 (where FFTC_x can be either FFTC_0 or FFTC_1).

Figure 7-29 shows the C6670 interrupt topology.

Figure 7-29 Interrupt Topology

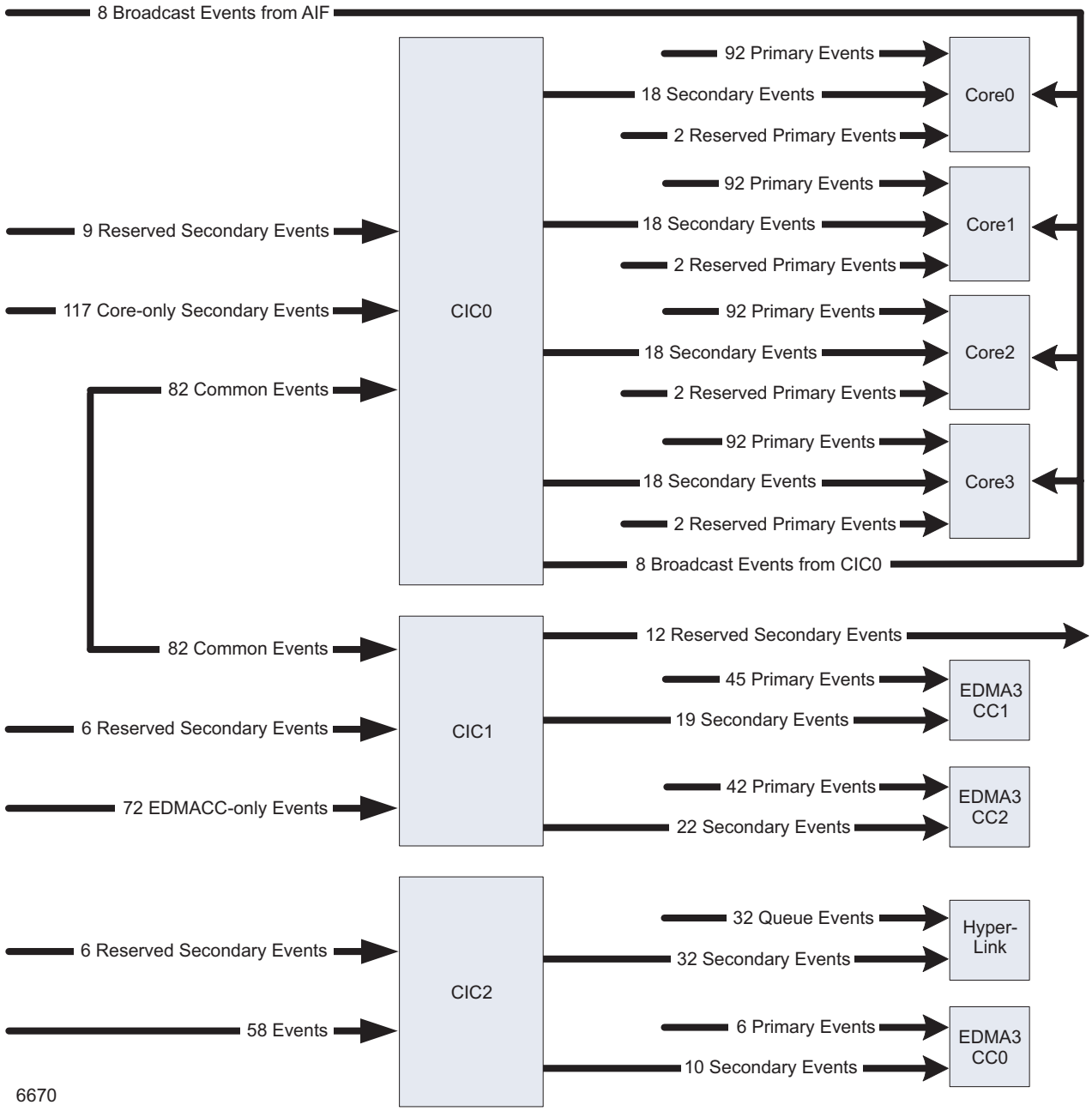


Table 7-38 shows the mapping of system events. For more information on the Interrupt Controller, see the *C66x CorePac User Guide* in2.9 "Related Documentation from Texas Instruments" on page 66.

Table 7-38 System Event Mapping — C66x CorePac Primary Interrupts (Part 1 of 4)

Event Number	Interrupt Event	Description
0	EVT0	Event combiner 0 output
1	EVT1	Event combiner 1 output
2	EVT2	Event combiner 2 output

Table 7-38 System Event Mapping — C66x CorePac Primary Interrupts (Part 2 of 4)

Event Number	Interrupt Event	Description
3	EVT3	Event combiner 3 output
4	TETBHFULLINTn[1]	TETB is half full
5	TETBFULLINTn[1]	TETB is full
6	TETBACQINTn[1]	Acquisition has been completed
7	TETBOVFLINTn[1]	Overflow condition interrupt
8	TETBUNFLINTn[1]	Underflow condition interrupt
9	EMU_DTDMA	Emulation interrupt for: 1. Host scan access 2. DTDMA transfer complete 3. AET interrupt
10	MSMC_mpf_errorn[4]	Memory protection fault indicators for local CorePac
11	EMU_RTDXRX	RTDX receive complete
12	EMU_RTDXTX	RTDX transmit complete
13	IDMA0	IDMA channel 0 interrupt
14	IDMA1	IDMA channel 1 interrupt
15	SEMERRn[2]	Semaphore error interrupt
16	SEMINTn[2]	Semaphore interrupt
17	PCIEXpress_MSI_INTn[3]	Message signaled interrupt mode
18	PCIEXpress_MSI_INTn+1[3]	Message signaled interrupt mode
19	RAC_A_INTn[8]	RAC_A interrupt
20	INTDST(*n*+16)[10]	SRIO interrupt
21	INTDST(*n*+20)[10]	SRIO interrupt
22	CIC0_OUT(64+0+10**n*)[7]	Interrupt Controller output
23	CIC0_OUT(64+1+10**n*)[7]	Interrupt Controller output
24	CIC0_OUT(64+2+10**n*)[7]	Interrupt Controller output
25	CIC0_OUT(64+3+10**n*)[7]	Interrupt Controller output
26	CIC0_OUT(64+4+10**n*)[7]	Interrupt Controller output
27	CIC0_OUT(64+5+10**n*)[7]	Interrupt Controller output
28	CIC0_OUT(64+6+10**n*)[7]	Interrupt Controller output
29	CIC0_OUT(64+7+10**n*)[7]	Interrupt Controller output
30	CIC0_OUT(64+8+10**n*)[7]	Interrupt Controller output
31	CIC0_OUT(64+9+10**n*)[7]	Interrupt Controller output
32	QM_INT_LOW_0	QM interrupt for 0~31 queues
33	QM_INT_LOW_1	QM interrupt for 32~63 queues
34	QM_INT_LOW_2	QM interrupt for 64~95 queues
35	QM_INT_LOW_3	QM interrupt for 96~127 queues
36	QM_INT_LOW_4	QM interrupt for 128~159 queues
37	QM_INT_LOW_5	QM interrupt for 160~191 queues
38	QM_INT_LOW_6	QM interrupt for 192~223 queues
39	QM_INT_LOW_7	QM interrupt for 224~255 queues
40	QM_INT_LOW_8	QM interrupt for 256~287 queues
41	QM_INT_LOW_9	QM interrupt for 288~319 queues
42	QM_INT_LOW_10	QM interrupt for 320~351 queues
43	QM_INT_LOW_11	QM interrupt for 352~383 queues
44	QM_INT_LOW_12	QM interrupt for 384~415 queues

Table 7-38 System Event Mapping — C66x CorePac Primary Interrupts (Part 3 of 4)

Event Number	Interrupt Event	Description
45	QM_INT_LOW_13	QM interrupt for 416~447 queues
46	QM_INT_LOW_14	QM interrupt for 448~479 queues
47	QM_INT_LOW_15	QM interrupt for 480~511 queues
48	QM_INT_HIGH_*n*[9]	QM interrupt for queue 704 + *n*[9]
49	QM_INT_HIGH_(*n*+4)[9]	QM interrupt for queue 708+*n*[9]
50	QM_INT_HIGH_(*n*+8)[9]	QM interrupt for queue 712+*n*[9]
51	QM_INT_HIGH_(*n*+12)[9]	QM interrupt for queue 716+*n*[9]
52	QM_INT_HIGH_(*n*+16)[9]	QM interrupt for queue 720+*n*[9]
53	QM_INT_HIGH_(*n*+20)[9]	QM interrupt for queue 724+*n*[9]
54	QM_INT_HIGH_(*n*+24)[9]	QM interrupt for queue 728+*n*[9]
55	QM_INT_HIGH_(*n*+28)[9]	QM interrupt for queue 732+*n*[9]
56	CIC0_OUT0	Interrupt Controller output
57	CIC0_OUT1	Interrupt Controller output
58	CIC0_OUT2	Interrupt Controller output
59	CIC0_OUT3	Interrupt Controller output
60	CIC0_OUT4	Interrupt Controller output
61	CIC0_OUT5	Interrupt Controller output
62	CIC0_OUT6	Interrupt Controller output
63	CIC0_OUT7	Interrupt Controller output
64	TINTL*n*[6]	Local Timer interrupt low
65	TINTH*n*[6]	Local Timer interrupt high
66	TINT4L	Timer 4 interrupt low
67	TINT4H	Timer 4 interrupt high
68	TINT5L	Timer 5 interrupt low
69	TINT5H	Timer 5 interrupt high
70	TINT6L	Timer 6 interrupt low
71	TINT6H	Timer 6 interrupt high
72	TINT7L	Timer 7 interrupt low
73	TINT7H	Timer 7 interrupt high
74	CIC0_OUT(8+16**n*)[7]	Interrupt Controller output
75	CIC0_OUT(9+16**n*)[7]	Interrupt Controller output
76	CIC0_OUT(10+16**n*)[7]	Interrupt Controller output
77	CIC0_OUT(11+16**n*)[7]	Interrupt Controller output
78	GPINT4	Local GPIO interrupt
79	GPINT5	Local GPIO interrupt
80	GPINT6	Local GPIO interrupt
81	GPINT7	Local GPIO interrupt
82	GPINT8	Local GPIO interrupt
83	GPINT9	Local GPIO interrupt
84	GPINT10	Local GPIO interrupt
85	GPINT11	Local GPIO interrupt
86	GPINT12	Local GPIO interrupt
87	GPINT13	Local GPIO interrupt
88	GPINT14	Local GPIO interrupt

Table 7-38 System Event Mapping — C66x CorePac Primary Interrupts (Part 4 of 4)

Event Number	Interrupt Event	Description
89	GPINT15	Local GPIO interrupt
90	IPC_LOCAL	Inter DSP interrupt from IPCGR*n*
91	GPINT*n*[5]	Local GPIO interrupt
92	CIC0_OUT(12+16**n*)[7]	Interrupt Controller output
93	CIC0_OUT(13+16**n*)[7]	Interrupt Controller output
94	CIC0_OUT(14+16**n*)[7]	Interrupt Controller output
95	CIC0_OUT(15+16**n*)[7]	Interrupt Controller output
96	INTERR	Dropped CPU interrupt event
97	EMC_IDMAERR	Invalid IDMA parameters
98	Reserved	
99	RAC_B_INT*n* [8]	RAC_B interrupt
100	EFIINTA	EFI interrupt from Side A
101	EFIINTB	EFI interrupt from Side B
102	AIF_SEVT0	AIF system event
103	AIF_SEVT1	AIF system event
104	AIF_SEVT2	AIF system event
105	AIF_SEVT3	AIF system event
106	AIF_SEVT4	AIF system event
107	AIF_SEVT5	AIF system event
108	AIF_SEVT6	AIF system event
109	AIF_SEVT7	AIF system event
110	MDMAERREVT	VbusM error event
111	Reserved	
112	EDMA3CC0_EDMACC_AETEVT	EDMA3CC0 AET event
113	PMC_ED	Single bit error detected during DMA read
114	EDMA3CC1_EDMACC_AETEVT	EDMA3CC1 AET event
115	EDMA3CC2_EDMACC_AETEVT	EDMA3CC2 AET event
116	UMC_ED1	Corrected bit error detected
117	UMC_ED2	Uncorrected bit error detected
118	PDC_INT	Power down sleep interrupt
119	SYS_CMPA	SYS CPU MP fault event
120	PMC_CMPA	CPU memory protection fault
121	PMC_DMPA	DMA memory protection fault
122	DMC_CMPA	CPU memory protection fault
123	DMC_DMPA	DMA memory protection fault
124	UMC_CMPA	CPU memory protection fault
125	UMC_DMPA	DMA memory protection fault
126	EMC_CMPA	CPU memory protection fault
127	EMC_BUSERR	Bus error interrupt
End of Table 7-38		

1. Core [n] will receive TETBHFULLINTn, TETBFULLINTn, TETBACQINTn, TETBOVFLINTn and TETBUNFLINTn.
2. Core [n] will receive SEMINTn and SEMERRn.
3. Core [n] will receive PCIEXpress_MSI_INTn and PCIEXpress_MSI_INTn+1.
4. Core [n] will receive MSMC_mpf_errorn.
5. Core [n] will receive GPINTn.

6. Core [n] will receive TINTLn and TINTHn.
7. For Core 0~3, it is CIC(interrupt number+17*n).
8. Core [n] will receive RACINTn.
9. n is core number.
10. Core [n] will receive INTDST(n+16) and INTDST(n+20).

Table 7-39 CIC0 Event Inputs — C66x CorePac Secondary Interrupts (Part 1 of 6)

Input Event# on CIC	System Interrupt	Description
0	EDMA3CC1 EDMACC_ERRINT	EDMA3CC1 error interrupt
1	EDMA3CC1 EDMACC_MPINT	EDMA3CC1 memory protection interrupt
2	EDMA3CC1 EDMATC_ERRINT0	EDMA3CC1 TC0 error interrupt
3	EDMA3CC1 EDMATC_ERRINT1	EDMA3CC1 TC1 error interrupt
4	EDMA3CC1 EDMATC_ERRINT2	EDMA3CC1 TC2 error interrupt
5	EDMA3CC1 EDMATC_ERRINT3	EDMA3CC1 TC3 error interrupt
6	EDMA3CC1 EDMACC_GINT	EDMA3CC1 GINT
7	Reserved	
8	EDMA3CC1 INT0	EDMA3CC1 individual completion interrupt
9	EDMA3CC1 INT1	EDMA3CC1 individual completion interrupt
10	EDMA3CC1INT2	EDMA3CC1 individual completion interrupt
11	EDMA3CC1 INT3	EDMA3CC1 individual completion interrupt
12	EDMA3CC1 INT4	EDMA3CC1 individual completion interrupt
13	EDMA3CC1 INT5	EDMA3CC1 individual completion interrupt
14	EDMA3CC1 INT6	EDMA3CC1 individual completion interrupt
15	EDMA3CC1 INT7	EDMA3CC1 individual completion interrupt
16	EDMA3CC2 EDMACC_ERRINT	EDMA3CC2 error interrupt
17	EDMA3CC2 EDMACC_MPINT	EDMA3CC2 memory protection interrupt
18	EDMA3CC2 EDMATC_ERRINT0	EDMA3CC2 TC0 error interrupt
19	EDMA3CC2 EDMATC_ERRINT1	EDMA3CC2 TC1 error interrupt
20	EDMA3CC2 EDMATC_ERRINT2	EDMA3CC2 TC2 error interrupt
21	EDMA3CC2 EDMATC_ERRINT3	EDMA3CC2 TC3 error interrupt
22	EDMA3CC2 EDMACC_GINT	EDMA3CC2 GINT
23	Reserved	
24	EDMA3CC2 INT0	EDMA3CC2 individual completion interrupt
25	EDMA3CC2 INT1	EDMA3CC2 individual completion interrupt
26	EDMA3CC2 INT2	EDMA3CC2 individual completion interrupt
27	EDMA3CC2 INT3	EDMA3CC2 individual completion interrupt
28	EDMA3CC2 INT4	EDMA3CC2 individual completion interrupt
29	EDMA3CC2 INT5	EDMA3CC2 individual completion interrupt
30	EDMA3CC2 INT6	EDMA3CC2 individual completion interrupt
31	EDMA3CC2 INT7	EDMA3CC2 individual completion interrupt
32	EDMA3CC0 EDMACC_ERRINT	EDMA3CC0 error interrupt
33	EDMA3CC0 EDMACC_MPINT	EDMA3CC0 memory protection interrupt
34	EDMA3CC0 EDMATC_ERRINT0	EDMA3CC0 TC0 error interrupt
35	EDMA3CC0 EDMATC_ERRINT1	EDMA3CC0 TC1 error interrupt
36	EDMA3CC0 EDMACC_GINT	EDMA3CC0 GINT
37	Reserved	
38	EDMA3CC0 INT0	EDMA3CC0 individual completion interrupt
39	EDMA3CC0 INT1	EDMA3CC0 individual completion interrupt

Table 7-39 CIC0 Event Inputs — C66x CorePac Secondary Interrupts (Part 2 of 6)

Input Event# on CIC	System Interrupt	Description
40	EDMA3CC0 INT2	EDMA3CC0 individual completion interrupt
41	EDMA3CC0 INT3	EDMA3CC0 individual completion interrupt
42	EDMA3CC0 INT4	EDMA3CC0 individual completion interrupt
43	EDMA3CC0 INT5	EDMA3CC0 individual completion interrupt
44	EDMA3CC0 INT6	EDMA3CC0 individual completion Interrupt
45	EDMA3CC0 INT7	EDMA3CC0 individual completion interrupt
46	Reserved	
47	Reserved	
48	PCIEXpress_ERR_INT	Protocol error interrupt
49	PCIEXpress_PM_INT	Power management interrupt
50	PCIEXpress_Legacy_INTA	Legacy interrupt mode
51	PCIEXpress_Legacy_INTB	Legacy interrupt mode
52	PCIEXpress_Legacy_INTC	Legacy interrupt mode
53	PCIEXpress_Legacy_INTD	Legacy interrupt mode
54	SPIINT0	SPI interrupt0
55	SPIINT1	SPI interrupt1
56	SPIXEVT	SPI transmit event
57	SPIREVT	SPI receive event
58	I2CINT	I^2C interrupt
59	I2CREVT	I^2C receive event
60	I2CXEVT	I^2C transmit event
61	Reserved	
62	Reserved	
63	TETBHFULLINT	TETB is half full
64	TETBFULLINT	TETB is full
65	TETBACQINT	Acquisition has been completed
66	TETBOVFLINT	Overflow condition occurred
67	TETBUNFLINT	Underflow condition occurred
68	mdio_link_intr0	Packet Accelerator subsystem MDIO interrupt
69	mdio_link_intr1	Packet Accelerator subsystem MDIO interrupt
70	mdio_user_intr0	Packet Accelerator subsystem MDIO interrupt
71	mdio_user_intr1	Packet Accelerator subsystem MDIO interrupt
72	misc_intr	Packet Accelerator subsystem misc Interrupt
73	Tracer_core_0_INTD	Tracer sliding time window interrupt for individual core
74	Tracer_core_1_INTD	Tracer sliding time window interrupt for individual core
75	Tracer_core_2_INTD	Tracer sliding time window interrupt for individual core
76	Tracer_core_3_INTD	Tracer sliding time window interrupt for individual core
77	Tracer_DDR_INTD	Tracer sliding time window interrupt for DDR3 EMIF1
78	Tracer_MSMC_0_INTD	Tracer sliding time window interrupt for MSMC SRAM bank0
79	Tracer_MSMC_1_INTD	Tracer sliding time window interrupt for MSMC SRAM bank1
80	Tracer_MSMC_2_INTD	Tracer sliding time window interrupt for MSMC SRAM bank2
81	Tracer_MSMC_3_INTD	Tracer sliding time window interrupt for MSMC SRAM bank3
82	Tracer_CFG_INTD	Tracer sliding time window interrupt for CFG0 SCR
83	Tracer_QM_SS_CFG_INTD	Tracer sliding time window interrupt for QM_SS CFG

Table 7-39 CIC0 Event Inputs — C66x CorePac Secondary Interrupts (Part 3 of 6)

Input Event# on CIC	System Interrupt	Description
84	Tracer_QM_SS_DMA_INTD	Tracer sliding time window interrupt for QM_SS slave
85	Tracer_SEM_INTD	Tracer sliding time window interrupt for Semaphore
86	PSC_ALLINT	Power & Sleep Controller Interrupt
87	MSMC_scrub_cerror	Correctable (1-bit) soft error detected during scrub cycle
88	BOOTCFG_INTD	Chip-level MMR Error Register
89	po_vcon_smpserr_intr	SmartReflex
90	MPU0_INTD (MPU0_ADDR_ERR_INT and MPU0_PROT_ERR_INT combined)	MPU0 addressing violation interrupt and protection violation interrupt.
91	BCP_ERROR0	BCP error 0
92	MPU1_INTD (MPU1_ADDR_ERR_INT and MPU1_PROT_ERR_INT combined)	MPU1 addressing violation interrupt and protection violation interrupt.
93	BCP_ERROR1	BCP error 1
94	MPU2_INTD (MPU2_ADDR_ERR_INT and MPU2_PROT_ERR_INT combined)	MPU2 addressing violation interrupt and protection violation interrupt.
95	BCP_ERROR2	BCP error 2
96	MPU3_INTD (MPU3_ADDR_ERR_INT and MPU3_PROT_ERR_INT combined)	MPU3 addressing violation interrupt and protection violation interrupt.
97	BCP_ERROR3	BCP error 3
98	MSMC_dedc_cerror	Correctable (1-bit) soft error detected on SRAM read
99	MSMC_dedc_nc_error	Non-correctable (2-bit) soft error detected on SRAM read
100	MSMC_scrub_nc_error	Non-correctable (2-bit) soft error detected during scrub cycle
101	Reserved	
102	MSMC_mpf_error8	Memory protection fault indicators for each system master PrivID
103	MSMC_mpf_error9	Memory protection fault indicators for each system master PrivID
104	MSMC_mpf_error10	Memory protection fault indicators for each system master PrivID
105	MSMC_mpf_error11	Memory protection fault indicators for each system master PrivID
106	MSMC_mpf_error12	Memory protection fault indicators for each system master PrivID
107	MSMC_mpf_error13	Memory protection fault indicators for each system master PrivID
108	MSMC_mpf_error14	Memory protection fault indicators for each system master PrivID
109	MSMC_mpf_error15	Memory protection fault indicators for each system master PrivID
110	DDR3_ERR	DDR3_EMIF Error Interrupt
111	vusr_int_o	HyperLink Interrupt
112	INTDST0	RapidIO Interrupt
113	INTDST1	RapidIO Interrupt
114	INTDST2	RapidIO Interrupt
115	INTDST3	RapidIO Interrupt
116	INTDST4	RapidIO Interrupt
117	INTDST5	RapidIO Interrupt
118	INTDST6	RapidIO Interrupt
119	INTDST7	RapidIO Interrupt
120	INTDST8	RapidIO Interrupt
121	INTDST9	RapidIO Interrupt
122	INTDST10	RapidIO Interrupt

Table 7-39 CIC0 Event Inputs — C66x CorePac Secondary Interrupts (Part 4 of 6)

Input Event# on CIC	System Interrupt	Description
123	INTDST11	RapidIO Interrupt
124	INTDST12	RapidIO Interrupt
125	INTDST13	RapidIO Interrupt
126	INTDST14	RapidIO interrupt
127	INTDST15	RapidIO interrupt
128	RACADEVENT0	RAC_A_debug event
129	RACADEVENT1	RAC_A_debug event
130	TAC_INTD	Error interrupt TACINT
131	TACDEVENT0	TAC debug event
132	TACDEVENT1	TAC debug event
133	AIF_INTD	AIF CPU error interrupt and AIF CPU alarm interrupt and starvation interrupt
134	QM_INT_PASS_TXQ_PEND_22	Queue Manager (Packet Accelerator) pend event
135	QM_INT_PASS_TXQ_PEND_23	Queue Manager (Packet Accelerator) pend event
136	QM_INT_PASS_TXQ_PEND_24	Queue Manager (Packet Accelerator) pend event
137	QM_INT_PASS_TXQ_PEND_25	Queue Manager (Packet Accelerator) pend event
138	QM_INT_PASS_TXQ_PEND_26	Queue Manager (Packet Accelerator) pend event
139	QM_INT_PASS_TXQ_PEND_27	Queue Manager (Packet Accelerator) pend event
140	QM_INT_PASS_TXQ_PEND_28	Queue Manager (Packet Accelerator) pend event
141	QM_INT_PASS_TXQ_PEND_29	Queue Manager (Packet Accelerator) pend event
142	QM_INT_PASS_TXQ_PEND_30	Queue Manager (Packet Accelerator) pend event
143	VCP0INT	Error interrupt
144	VCP1INT	Error interrupt
145	VCP2INT	Error interrupt
146	VCP3INT	Error interrupt
147	VCP0REVT	Receive event
148	VCP0XEVT	Transmit event
149	VCP1REVT	Receive event
150	VCP1XEVT	Transmit event
151	VCP2REVT	Receive event
152	VCP2XEVT	Transmit event
153	VCP3REVT	Receive event
154	VCP3XEVT	Transmit event
155	TCP3D_A_INTD	TCP3d_A error interrupt TCP3DINT0 and TCP3DINT1
156	TCP3D_B_INTD	TCP3d_B error interrupt TCP3DINT0 and TCP3DINT1
157	TCP3D_AREVT0	TCP3d_A receive event0
158	TCP3D_AREVT1	TCP3d_A receive event1
159	TCP3E_INTD	Error interrupt TCP3EINT
160	TCP3EREVT	TCP3e read event
161	TCP3EWEVT	TCP3e write event
162	TCP3D_BREVT0	TCP3d_B receive event0
163	TCP3D_BREVT1	TCP3d_B receive event1
164	UARTINT	UART interrupt
165	URXEVT	UART receive event
166	UTXEVT	UART transmit event

Table 7-39 CIC0 Event Inputs — C66x CorePac Secondary Interrupts (Part 5 of 6)

Input Event# on CIC	System Interrupt	Description
167	Tracer_RAC_INTD	Tracer sliding time window interrupt for RAC
168	Tracer_RAC_FE_INTD	Tracer sliding time window interrupt for RAC_FE
169	Tracer_TAC_INTD	Tracer sliding time window interrupt for TAC
170	MSMC_mpf_error4	Memory protection fault indicators for each system master PrivID
171	MSMC_mpf_error5	Memory protection fault indicators for each system master PrivID
172	MSMC_mpf_error6	Memory protection fault indicators for each system master PrivID
173	MSMC_mpf_error7	Memory protection fault indicators for each system master PrivID
174	MPU4_INTD (MPU4_ADDR_ERR_INT and MPU4_PROT_ERR_INT combined)	MPU4 addressing violation interrupt and protection violation interrupt.
175	QM_INT_PASS_TXQ_PEND_31	Queue Manager (Packet Accelerator) pend event
176	QM_INT_CDMA_0	QM interrupt for CDMA starvation
177	QM_INT_CDMA_1	QM interrupt for CDMA starvation
178	RapidIO_INT_CDMA_0	RapidIO interrupt for CDMA starvation
179	PASS_INT_CDMA_0	PASS interrupt for CDMA starvation
180	TCP3D_C_ERROR MPU5_INTD (MPU5_ADDR_ERR_INT and MPU5_PROT_ERR_INT combined)	TCP3D_C_Error event MPU5 Addressing violation interrupt and Protection violation interrupt.
181	SmartReflex_intrreq0	SmartReflex sensor interrupt
182	SmartReflex_intrreq1	SmartReflex sensor interrupt
183	SmartReflex_intrreq2	SmartReflex sensor interrupt
184	SmartReflex_intrreq3	SmartReflex sensor interrupt
185	VPNoSMPSAck	VPVOLTUPDATE has been asserted but SMPS has not been responded to in a defined time interval
186	VPEqValue	$\overline{\text{SRSINTERUPT}}$ is asserted, but the new voltage is not different from the current SMPS voltage
187	VPMaxVdd	The new voltage required is equal to or greater than MaxVdd.
188	VPMinVdd	The new voltage required is equal to or less than MinVdd.
189	VPINIDLE	Indicating that the FSM of voltage processor is in idle.
190	VPOPPChangeDone	Indicating that the average frequency error is within the desired limit.
191	Reserved	
192	FFTC_A_INTD0	FFTC_A error event and FFTC_A debug event
193	FFTC_A_INTD1	FFTC_A error event and FFTC_A debug event
194	FFTC_A_INTD2	FFTC_A error event and FFTC_A debug event
195	FFTC_A_INTD3	FFTC_A error event and FFTC_A debug event
196	FFTC_B_INTD0	FFTC_B error event and FFTC_B debug event
197	FFTC_B_INTD1	FFTC_B error event and FFTC_B debug event
198	FFTC_B_INTD2	FFTC_B error event and FFTC_B debug event
199	FFTC_B_INTD3	FFTC_B error event and FFTC_B debug event
200	RACBDEVENT0	RAC_B_debug Event
201	RACBDEVENT1	RAC_B_debug Event
202	TCP3D_C_REVT0	TCP3d_C receive event0
203	TCP3D_C_REVT1	TCP3d_C receive event1
204	FFTC_C_ERROR0	FFTC_C Error event and FFTC_C debug event
205	FFTC_C_ERROR1	FFTC_C Error event and FFTC_C debug event

Table 7-39 CIC0 Event Inputs — C66x CorePac Secondary Interrupts (Part 6 of 6)

Input Event# on CIC	System Interrupt	Description
206	FFTC_C_ERROR2	FFTC_C Error event and FFTC_C debug event
207	FFTC_C_ERROR3	FFTC_C Error event and FFTC_C debug event
End of Table 7-39		

Table 7-40 CIC1 Event Inputs (Secondary Events for EDMA3CC1 and EDMA3CC2) (Part 1 of 4)

Input Event # on CIC	System Interrupt	Description
0	GPINT8	GPIO interrupt
1	GPINT9	GPIO interrupt
2	GPINT10	GPIO interrupt
3	GPINT11	GPIO interrupt
4	GPINT12	GPIO interrupt
5	GPINT13	GPIO interrupt
6	GPINT14	GPIO interrupt
7	GPINT15	GPIO interrupt
8	TETBHFULLINT	TETB is half full
9	TETBFULLINT	TETB is full
10	TETBACQINT	Acquisition has been completed
11	TETBHFULLINT0	TETB is half full
12	TETBFULLINT0	TETB is full
13	TETBACQINT0	Acquisition has been completed
14	TETBHFULLINT1	TETB is half full
15	TETBFULLINT1	TETB is full
16	TETBACQINT1	Acquisition has been completed
17	TETBHFULLINT2	TETB is half full
18	TETBFULLINT2	TETB is full
19	TETBACQINT2	Acquisition has been completed
20	TETBHFULLINT3	TETB is half full
21	TETBFULLINT3	TETB is full
22	TETBACQINT3	Acquisition has been completed
23	Reserved	
24	QM_INT_HIGH_16	QM interrupt
25	QM_INT_HIGH_17	QM interrupt
26	QM_INT_HIGH_18	QM interrupt
27	QM_INT_HIGH_19	QM interrupt
28	QM_INT_HIGH_20	QM interrupt
29	QM_INT_HIGH_21	QM interrupt
30	QM_INT_HIGH_22	QM interrupt
31	QM_INT_HIGH_23	QM interrupt
32	QM_INT_HIGH_24	QM interrupt
33	QM_INT_HIGH_25	QM interrupt
34	QM_INT_HIGH_26	QM interrupt
35	QM_INT_HIGH_27	QM interrupt
36	QM_INT_HIGH_28	QM interrupt

Table 7-40 CIC1 Event Inputs (Secondary Events for EDMA3CC1 and EDMA3CC2) (Part 2 of 4)

Input Event # on CIC	System Interrupt	Description
37	QM_INT_HIGH_29	QM interrupt
38	QM_INT_HIGH_30	QM interrupt
39	QM_INT_HIGH_31	QM interrupt
40	MDIO_LINK_INTR0	PASS_MDIO interrupt
41	MDIO_LINK_INTR1	PASS_MDIO interrupt
42	MDIO_USER_INTR0	PASS_MDIO interrupt
43	MDIO_USER_INTR1	PASS_MDIO interrupt
44	MISC_INTR	PASS_MISC interrupt
45	TRACER_CORE_0_INTD	Tracer sliding time window interrupt for individual core
46	TRACER_CORE_1_INTD	Tracer sliding time window interrupt for individual core
47	TRACER_CORE_2_INTD	Tracer sliding time window interrupt for individual core
48	TRACER_CORE_3_INTD	Tracer sliding time window interrupt for individual core
49	TRACER_DDR_INTD	Tracer sliding time window interrupt for DDR3 EMIF1
50	TRACER_MSMC_0_INTD	Tracer sliding time window interrupt for MSMC SRAM bank0
51	TRACER_MSMC_1_INTD	Tracer sliding time window interrupt for MSMC SRAM bank1
52	TRACER_MSMC_2_INTD	Tracer sliding time window interrupt for MSMC SRAM bank2
53	TRACER_MSMC_3_INTD	Tracer sliding time window interrupt for MSMC SRAM bank3
54	TRACER_CFG_INTD	Tracer sliding time window interrupt for CFG0 SCR
55	TRACER_QM_SS_CFG_INTD	Tracer sliding time window interrupt for QM_SS CFG
56	TRACER_QM_SS_DMA_INTD	Tracer sliding time window interrupt for QM_SS slave port
57	TRACER_SEM_INTD	Tracer sliding time window interrupt for Semaphore
58	SEMERR0	Semaphore interrupt
59	SEMERR1	Semaphore interrupt
60	SEMERR2	Semaphore interrupt
61	SEMERR3	Semaphore interrupt
62	BOOTCFG_INTD	Chip-level MMR interrupt
63	PASS_INT_CDMA_0	PASS Interrupt for CDMA starvation
64	MPU0_INTD (MPU0_ADDR_ERR_INT and MPU0_PROT_ERR_INT combined)	MPU0 addressing violation interrupt and protection violation interrupt.
65	MSMC_SCRUB_CERROR	Correctable (1-bit) soft error detected during scrub cycle
66	MPU1_INTD (MPU1_ADDR_ERR_INT and MPU1_PROT_ERR_INT combined)	MPU1 addressing violation interrupt and protection violation interrupt.
67	RapidIO_INT_CDMA_0	RapidIO interrupt for CDMA starvation
68	MPU2_INTD (MPU2_ADDR_ERR_INT and MPU2_PROT_ERR_INT combined)	MPU2 addressing violation interrupt and protection violation interrupt.
69	QM_INT_CDMA_0	QM Interrupt for CDMA starvation
70	MPU3_INTD (MPU3_ADDR_ERR_INT and MPU3_PROT_ERR_INT combined)	MPU3 addressing violation interrupt and protection violation interrupt.
71	QM_INT_CDMA_1	QM interrupt for CDMA starvation
72	MSMC_DEDC_CERROR	Correctable (1-bit) soft error detected on SRAM read
73	MSMC_DEDC_NC_ERROR	Non-correctable (2-bit) soft error detected on SRAM read
74	MSMC_SCRUB_NC_ERROR	Non-correctable (2-bit) soft error detected during scrub cycle
75	Reserved	

Table 7-40 CIC1 Event Inputs (Secondary Events for EDMA3CC1 and EDMA3CC2) (Part 3 of 4)

Input Event # on CIC	System Interrupt	Description
76	MSMC_MPF_ERROR0	Memory protection fault indicators for each system master PrivID
77	MSMC_MPF_ERROR1	Memory protection fault indicators for each system master PrivID
78	MSMC_MPF_ERROR2	Memory protection fault indicators for each system master PrivID
79	MSMC_MPF_ERROR3	Memory protection fault indicators for each system master PrivID
80	MSMC_MPF_ERROR4	Memory protection fault indicators for each system master PrivID
81	MSMC_MPF_ERROR5	Memory protection fault indicators for each system master PrivID
82	MSMC_MPF_ERROR6	Memory protection fault indicators for each system master PrivID
83	MSMC_MPF_ERROR7	Memory protection fault indicators for each system master PrivID
84	MSMC_MPF_ERROR8	Memory protection fault indicators for each system master PrivID
85	MSMC_MPF_ERROR9	Memory protection fault indicators for each system master PrivID
86	MSMC_MPF_ERROR10	Memory protection fault indicators for each system master PrivID
87	MSMC_MPF_ERROR11	Memory protection fault indicators for each system master PrivID
88	MSMC_MPF_ERROR12	Memory protection fault indicators for each system master PrivID
89	MSMC_MPF_ERROR13	Memory protection fault indicators for each system master PrivID
90	MSMC_MPF_ERROR14	Memory protection fault indicators for each system master PrivID
91	MSMC_MPF_ERROR15	Memory protection fault indicators for each system master PrivID
92	Reserved	
93	INTDST0	RapidIO interrupt
94	INTDST1	RapidIO interrupt
95	INTDST2	RapidIO interrupt
96	INTDST3	RapidIO interrupt
97	INTDST4	RapidIO interrupt
98	INTDST5	RapidIO interrupt
99	INTDST6	RapidIO interrupt
100	INTDST7	RapidIO interrupt
101	INTDST8	RapidIO interrupt
102	INTDST9	RapidIO interrupt
103	INTDST10	RapidIO interrupt
104	INTDST11	RapidIO interrupt
105	INTDST12	RapidIO interrupt
106	INTDST13	RapidIO interrupt
107	INTDST14	RapidIO interrupt
108	INTDST15	RapidIO interrupt
109	INTDST16	RapidIO interrupt
110	INTDST17	RapidIO interrupt
111	INTDST18	RapidIO interrupt
112	INTDST19	RapidIO interrupt
113	INTDST20	RapidIO interrupt
114	INTDST21	RapidIO interrupt
115	INTDST22	RapidIO interrupt
116	INTDST23	RapidIO interrupt
117	AIF_INTD	AIF CPU error interrupt and AIF CPU alarm interrupt and starvation interrupt
118	Reserved	
119	VCPAINT	Error interrupt

Table 7-40 CIC1 Event Inputs (Secondary Events for EDMA3CC1 and EDMA3CC2) (Part 4 of 4)

Input Event # on CIC	System Interrupt	Description
120	VCPBINT	Error interrupt
121	VCPCINT	Error interrupt
122	VCPDINT	Error interrupt
123	TCP3D_A_INTD	Error interrupt TCP3DINT0 and TCP3DINT1
124	TCP3D_B_INTD	Error interrupt TCP3DINT0 and TCP3DINT1
125	TCP3E_INTD	Error interrupt TCP3EINT
126	FFTC_B_INTD0	FFTC_B error event and FFTC_B debug event
127	FFTC_B_INTD1	FFTC_B error event and FFTC_B debug event
128	GPINT4	GPIO interrupt
129	GPINT5	GPIO interrupt
130	GPINT6	GPIO interrupt
131	GPINT7	GPIO interrupt
132	TRACER_RAC_INTD	Tracer sliding time window interrupt for RAC
133	TRACER_RAC_FE_INTD	Tracer sliding time window interrupt for RAC_FE
134	TRACER_TAC_INTD	Tracer sliding time window interrupt for TAC
135	MPU4_INTD	MPU4 addressing violation interrupt and protection violation interrupt.
136	Reserved	
137	QM_INT_HIGH_0	QM interrupt
138	QM_INT_HIGH_1	QM interrupt
139	QM_INT_HIGH_2	QM interrupt
140	QM_INT_HIGH_3	QM interrupt
141	QM_INT_HIGH_4	QM interrupt
142	QM_INT_HIGH_5	QM interrupt
143	QM_INT_HIGH_6	QM interrupt
144	QM_INT_HIGH_7	QM interrupt
145	QM_INT_HIGH_8	QM interrupt
146	QM_INT_HIGH_9	QM interrupt
147	QM_INT_HIGH_10	QM interrupt
148	QM_INT_HIGH_11	QM interrupt
149	QM_INT_HIGH_12	QM interrupt
150	QM_INT_HIGH_13	QM interrupt
151	QM_INT_HIGH_14	QM interrupt
152	QM_INT_HIGH_15	QM interrupt
153	FFTC_A_INTD0	FFTC_A error event and FFTC_A debug event
154	FFTC_A_INTD1	FFTC_A error event and FFTC_A debug event
155	FFTC_A_INTD2	FFTC_A error event and FFTC_A debug event
156	FFTC_A_INTD3	FFTC_A error event and FFTC_A debug event
157	FFTC_B_INTD2	FFTC_B error event and FFTC_B debug event
158	FFTC_B_INTD3	FFTC_B error event and FFTC_B debug event
159	Reserved	Reserved inputs
End of Table 7-40		

Table 7-41 CIC2 Event Inputs (Secondary Events for EDMA3CC0 and HyperLink) (Part 1 of 2)

Input Event # on CIC	System Interrupt	Description
0	GPINT0	GPIO interrupt
1	GPINT1	GPIO interrupt
2	GPINT2	GPIO interrupt
3	GPINT3	GPIO interrupt
4	GPINT4	GPIO interrupt
5	GPINT5	GPIO interrupt
6	GPINT6	GPIO interrupt
7	GPINT7	GPIO interrupt
8	GPINT8	GPIO interrupt
9	GPINT9	GPIO interrupt
10	GPINT10	GPIO interrupt
11	GPINT11	GPIO interrupt
12	GPINT12	GPIO interrupt
13	GPINT13	GPIO interrupt
14	GPINT14	GPIO interrupt
15	GPINT15	GPIO interrupt
16	TETBHFULLINT	System TETB is half full
17	TETBFULLINT	System TETB is full
18	TETBACQINT	System acquisition has been completed
19	TETBHFULLINT0	TETB0 is half full
20	TETBFULLINT0	TETB0 is full
21	TETBACQINT0	TETB0 acquisition has been completed
22	TETBHFULLINT1	TETB1 is half full
23	TETBFULLINT1	TETB1 is full
24	TETBACQINT1	TETB1 acquisition has been completed
25	TETBHFULLINT2	TETB2 is half full
26	TETBFULLINT2	TETB2 is full
27	TETBACQINT2	TETB2 acquisition has been completed
28	TETBHFULLINT3	TETB3 is half full
29	TETBFULLINT3	TETB3 is full
30	TETBACQINT3	TETB3 acquisition has been completed
31	TRACER_CORE_0_INTD	Tracer sliding time window interrupt for individual core
32	TRACER_CORE_1_INTD	Tracer sliding time window interrupt for individual core
33	TRACER_CORE_2_INTD	Tracer sliding time window interrupt for individual core
34	TRACER_CORE_3_INTD	Tracer sliding time window interrupt for individual core
35	TRACER_DDR_INTD	Tracer sliding time window interrupt for DDR3 EMIF1
36	TRACER_MSMC_0_INTD	Tracer sliding time window interrupt for MSMC SRAM bank0
37	TRACER_MSMC_1_INTD	Tracer sliding time window interrupt for MSMC SRAM bank1
38	TRACER_MSMC_2_INTD	Tracer sliding time window interrupt for MSMC SRAM bank2
39	TRACER_MSMC_3_INTD	Tracer sliding time window interrupt for MSMC SRAM bank3
40	TRACER_CFG_INTD	Tracer sliding time window interrupt for CFG0 SCR
41	TRACER_QM_SS_CFG_INTD	Tracer sliding time window interrupt for QM_SS CFG
42	TRACER_QM_SS_DMA_INTD	Tracer sliding time window interrupt for QM_SS slave port
43	TRACER_SEM_INTD	Tracer sliding time window interrupt for Semaphore

Table 7-41 CIC2 Event Inputs (Secondary Events for EDMA3CC0 and HyperLink) (Part 2 of 2)

Input Event # on CIC	System Interrupt	Description
44	VUSR_INT_O	HyperLink interrupt
45	TRACER_RAC_INTD	Tracer sliding time window interrupt for RAC
46	TRACER_RAC_FE_INTD	Tracer sliding time window interrupt for RAC_FE
47	TRACER_TAC_INTD	Tracer sliding time window interrupt for TAC
48	TCP3D_C_ERROR MPU5_INTD (MPU5_ADDR_ERR_INT and MPU5_PROT_ERR_INT combined)	TCP3D_C Error Event MPU5 Addressing violation interrupt and Protection violation interrupt.
49	TINT4L	Timer64_4 interrupt low
50	TINT4H	Timer64_4 interrupt high
51	TINT5L	Timer64_5 interrupt low
52	TINT5H	timer64_5 interrupt high
53	TINT6L	Timer64_6 interrupt low
54	TINT6H	Timer64_6 interrupt high
55	TINT7L	Timer64_7 interrupt low
56	TINT7H	Timer64_7 interrupt high
57	Reserved	
58	Reserved	
59	Reserved	
60	Reserved	
61	DDR3_ERR	DDR3 EMIF error interrupt
62	Reserved	
63	Reserved	
End of Table 7-41		

7.9.2 CIC Registers

This section includes the CIC memory map information and registers.

7.9.2.1 CIC0 Register Map

Table 7-42 CIC0 Registers (Part 1 of 4)

Address Offset	Register Mnemonic	Register Name
0x0	REVISION_REG	Revision Register
0x4	Reserved	
0xc	Reserved	
0x10	GLOBAL_ENABLE_HINT_REG	Global Host Int Enable Register
0x20	STATUS_SET_INDEX_REG	Status Set Index Register
0x24	STATUS_CLR_INDEX_REG	Status Clear Index Register
0x28	ENABLE_SET_INDEX_REG	Enable Set Index Register
0x2c	ENABLE_CLR_INDEX_REG	Enable Clear Index Register
0x34	HINT_ENABLE_SET_INDEX_REG	Host Int Enable Set Index Register
0x38	HINT_ENABLE_CLR_INDEX_REG	Host Int Enable Clear Index Register
0x200	RAW_STATUS_REG0	Raw Status Register 0
0x204	RAW_STATUS_REG1	Raw Status Register 1

Table 7-42 CIC0 Registers (Part 2 of 4)

Address Offset	Register Mnemonic	Register Name
0x208	RAW_STATUS_REG2	Raw Status Register 2
0x20c	RAW_STATUS_REG3	Raw Status Register 3
0x210	RAW_STATUS_REG4	Raw Status Register 4
0x214	RAW_STATUS_REG5	Raw Status Register 5
0x218	RAW_STATUS_REG6	Raw Status Register 6
0x280	ENA_STATUS_REG0	Enabled Status Register 0
0x284	ENA_STATUS_REG1	Enabled Status Register 1
0x288	ENA_STATUS_REG2	Enabled Status Register 2
0x28c	ENA_STATUS_REG3	Enabled Status Register 3
0x290	ENA_STATUS_REG4	Enabled Status Register 4
0x294	ENA_STATUS_REG5	Enabled Status Register 5
0x298	ENA_STATUS_REG6	Enabled Status Register 6
0x300	ENABLE_REG0	Enable Register 0
0x304	ENABLE_REG1	Enable Register 1
0x308	ENABLE_REG2	Enable Register 2
0x30c	ENABLE_REG3	Enable Register 3
0x310	ENABLE_REG4	Enable Register 4
0x314	ENABLE_REG5	Enable Register 5
0x318	ENABLE_REG6	Enable Register 6
0x380	ENABLE_CLR_REG0	Enable Clear Register 0
0x384	ENABLE_CLR_REG1	Enable Clear Register 1
0x388	ENABLE_CLR_REG2	Enable Clear Register 2
0x38c	ENABLE_CLR_REG3	Enable Clear Register 3
0x390	ENABLE_CLR_REG4	Enable Clear Register 4
0x394	ENABLE_CLR_REG5	Enable Clear Register 5
0x398	ENABLE_CLR_REG6	Enable Clear Register 6
0x400	CH_MAP_REG0	Interrupt Channel Map Register for 0 to 0+3
0x404	CH_MAP_REG1	Interrupt Channel Map Register for 4 to 4+3
0x408	CH_MAP_REG2	Interrupt Channel Map Register for 8 to 8+3
0x40c	CH_MAP_REG3	Interrupt Channel Map Register for 12 to 12+3
0x410	CH_MAP_REG4	Interrupt Channel Map Register for 16 to 16+3
0x414	CH_MAP_REG5	Interrupt Channel Map Register for 20 to 20+3
0x418	CH_MAP_REG6	Interrupt Channel Map Register for 24 to 24+3
0x41c	CH_MAP_REG7	Interrupt Channel Map Register for 28 to 28+3
0x420	CH_MAP_REG8	Interrupt Channel Map Register for 32 to 32+3
0x424	CH_MAP_REG9	Interrupt Channel Map Register for 36 to 36+3
0x428	CH_MAP_REG10	Interrupt Channel Map Register for 40 to 40+3
0x42c	CH_MAP_REG11	Interrupt Channel Map Register for 44 to 44+3
0x430	CH_MAP_REG12	Interrupt Channel Map Register for 48 to 48+3
0x434	CH_MAP_REG13	Interrupt Channel Map Register for 52 to 52+3
0x438	CH_MAP_REG14	Interrupt Channel Map Register for 56 to 56+3
0x43c	CH_MAP_REG15	Interrupt Channel Map Register for 60 to 60+3
0x440	CH_MAP_REG16	Interrupt Channel Map Register for 64 to 64+3
0x444	CH_MAP_REG17	Interrupt Channel Map Register for 68 to 68+3

Table 7-42 CIC0 Registers (Part 3 of 4)

Address Offset	Register Mnemonic	Register Name
0x448	CH_MAP_REG18	Interrupt Channel Map Register for 72 to 72+3
0x44c	CH_MAP_REG19	Interrupt Channel Map Register for 76 to 76+3
0x450	CH_MAP_REG20	Interrupt Channel Map Register for 80 to 80+3
0x454	CH_MAP_REG21	Interrupt Channel Map Register for 84 to 84+3
0x458	CH_MAP_REG22	Interrupt Channel Map Register for 88 to 88+3
0x45c	CH_MAP_REG23	Interrupt Channel Map Register for 92 to 92+3
0x460	CH_MAP_REG24	Interrupt Channel Map Register for 96 to 96+3
0x464	CH_MAP_REG25	Interrupt Channel Map Register for 100 to 100+3
0x468	CH_MAP_REG26	Interrupt Channel Map Register for 104 to 104+3
0x46c	CH_MAP_REG27	Interrupt Channel Map Register for 108 to 108+3
0x470	CH_MAP_REG28	Interrupt Channel Map Register for 112 to 112+3
0x474	CH_MAP_REG29	Interrupt Channel Map Register for 116 to 116+3
0x478	CH_MAP_REG30	Interrupt Channel Map Register for 120 to 120+3
0x47c	CH_MAP_REG31	Interrupt Channel Map Register for 124 to 124+3
0x480	CH_MAP_REG32	Interrupt Channel Map Register for 128 to 128+3
0x484	CH_MAP_REG33	Interrupt Channel Map Register for 132 to 132+3
0x488	CH_MAP_REG34	Interrupt Channel Map Register for 136 to 136+3
0x48c	CH_MAP_REG35	Interrupt Channel Map Register for 140 to 140+3
0x490	CH_MAP_REG36	Interrupt Channel Map Register for 144 to 144+3
0x494	CH_MAP_REG37	Interrupt Channel Map Register for 148 to 148+3
0x498	CH_MAP_REG38	Interrupt Channel Map Register for 152 to 152+3
0x49c	CH_MAP_REG39	Interrupt Channel Map Register for 156 to 156+3
0x4a0	CH_MAP_REG40	Interrupt Channel Map Register for 160 to 160+3
0x4a4	CH_MAP_REG41	Interrupt Channel Map Register for 164 to 164+3
0x4a8	CH_MAP_REG42	Interrupt Channel Map Register for 168 to 168+3
0x4ac	CH_MAP_REG43	Interrupt Channel Map Register for 172 to 172+3
0x4b0	CH_MAP_REG44	Interrupt Channel Map Register for 176 to 176+3
0x4b4	CH_MAP_REG45	Interrupt Channel Map Register for 180 to 180+3
0x4b8	CH_MAP_REG46	Interrupt Channel Map Register for 184 to 184+3
0x4bc	CH_MAP_REG47	Interrupt Channel Map Register for 188 to 188+3
0x4c0	CH_MAP_REG48	Interrupt Channel Map Register for 192 to 192+3
0x4c4	CH_MAP_REG49	Interrupt Channel Map Register for 196 to 196+3
0x4c8	CH_MAP_REG50	Interrupt Channel Map Register for 200 to 200+3
0x4cc	CH_MAP_REG51	Interrupt Channel Map Register for 204 to 204+3
0x800	HINT_MAP_REG0	Host Interrupt Map Register for 0 to 0+3
0x804	HINT_MAP_REG1	Host Interrupt Map Register for 4 to 4+3
0x808	HINT_MAP_REG2	Host Interrupt Map Register for 8 to 8+3
0x80c	HINT_MAP_REG3	Host Interrupt Map Register for 12 to 12+3
0x810	HINT_MAP_REG4	Host Interrupt Map Register for 16 to 16+3
0x814	HINT_MAP_REG5	Host Interrupt Map Register for 20 to 20+3
0x818	HINT_MAP_REG6	Host Interrupt Map Register for 24 to 24+3
0x81c	HINT_MAP_REG7	Host Interrupt Map Register for 28 to 28+3
0x820	HINT_MAP_REG8	Host Interrupt Map Register for 32 to 32+3
0x824	HINT_MAP_REG9	Host Interrupt Map Register for 36 to 36+3

Table 7-42 CIC0 Registers (Part 4 of 4)

Address Offset	Register Mnemonic	Register Name
0x828	HINT_MAP_REG10	Host Interrupt Map Register for 40 to 40+3
0x82c	HINT_MAP_REG11	Host Interrupt Map Register for 44 to 44+3
0x830	HINT_MAP_REG12	Host Interrupt Map Register for 48 to 48+3
0x834	HINT_MAP_REG13	Host Interrupt Map Register for 52 to 52+3
0x838	HINT_MAP_REG14	Host Interrupt Map Register for 56 to 56+3
0x83c	HINT_MAP_REG15	Host Interrupt Map Register for 60 to 60+3
0x840	HINT_MAP_REG16	Host Interrupt Map Register for 64 to 64+3
0x844	HINT_MAP_REG17	Host Interrupt Map Register for 68 to 68+3
0x848	HINT_MAP_REG18	Host Interrupt Map Register for 72 to 72+3
0x84c	HINT_MAP_REG19	Host Interrupt Map Register for 76 to 76+3
0x1500	ENABLE_HINT_REG0	Host Int Enable Register 0
0x1504	ENABLE_HINT_REG1	Host Int Enable Register 1
0x1508	ENABLE_HINT_REG2	Host Int Enable Register 2
End of Table 7-42		

7.9.2.2 CIC1 Register Map

Table 7-43 CIC1 Registers (Part 1 of 3)

Address Offset	Register Mnemonic	Register Name
0x0	REVISION_REG	Revision Register
0x10	GLOBAL_ENABLE_HINT_REG	Global Host Int Enable Register
0x20	STATUS_SET_INDEX_REG	Status Set Index Register
0x24	STATUS_CLR_INDEX_REG	Status Clear Index Register
0x28	ENABLE_SET_INDEX_REG	Enable Set Index Register
0x2c	ENABLE_CLR_INDEX_REG	Enable Clear Index Register
0x34	HINT_ENABLE_SET_INDEX_REG	Host Int Enable Set Index Register
0x38	HINT_ENABLE_CLR_INDEX_REG	Host Int Enable Clear Index Register
0x200	RAW_STATUS_REG0	Raw Status Register 0
0x204	RAW_STATUS_REG1	Raw Status Register 1
0x208	RAW_STATUS_REG2	Raw Status Register 2
0x20c	RAW_STATUS_REG3	Raw Status Register 3
0x210	RAW_STATUS_REG4	Raw Status Register 4
0x280	ENA_STATUS_REG0	Enabled Status Register 0
0x284	ENA_STATUS_REG1	Enabled Status Register 1
0x288	ENA_STATUS_REG2	Enabled Status Register 2
0x28c	ENA_STATUS_REG3	Enabled Status Register 3
0x290	ENA_STATUS_REG4	Enabled Status Register 4
0x300	ENABLE_REG0	Enable Register 0
0x304	ENABLE_REG1	Enable Register 1
0x308	ENABLE_REG2	Enable Register 2
0x30c	ENABLE_REG3	Enable Register 3
0x310	ENABLE_REG4	Enable Register 4
0x380	ENABLE_CLR_REG0	Enable Clear Register 0
0x384	ENABLE_CLR_REG1	Enable Clear Register 1

Table 7-43 CIC1 Registers (Part 2 of 3)

Address Offset	Register Mnemonic	Register Name
0x388	ENABLE_CLR_REG2	Enable Clear Register 2
0x38c	ENABLE_CLR_REG3	Enable Clear Register 3
0x390	ENABLE_CLR_REG4	Enable Clear Register 4
0x400	CH_MAP_REG0	Interrupt Channel Map Register for 0 to 0+3
0x404	CH_MAP_REG1	Interrupt Channel Map Register for 4 to 4+3
0x408	CH_MAP_REG2	Interrupt Channel Map Register for 8 to 8+3
0x40c	CH_MAP_REG3	Interrupt Channel Map Register for 12 to 12+3
0x410	CH_MAP_REG4	Interrupt Channel Map Register for 16 to 16+3
0x414	CH_MAP_REG5	Interrupt Channel Map Register for 20 to 20+3
0x418	CH_MAP_REG6	Interrupt Channel Map Register for 24 to 24+3
0x41c	CH_MAP_REG7	Interrupt Channel Map Register for 28 to 28+3
0x420	CH_MAP_REG8	Interrupt Channel Map Register for 32 to 32+3
0x424	CH_MAP_REG9	Interrupt Channel Map Register for 36 to 36+3
0x428	CH_MAP_REG10	Interrupt Channel Map Register for 40 to 40+3
0x42c	CH_MAP_REG11	Interrupt Channel Map Register for 44 to 44+3
0x430	CH_MAP_REG12	Interrupt Channel Map Register for 48 to 48+3
0x434	CH_MAP_REG13	Interrupt Channel Map Register for 52 to 52+3
0x438	CH_MAP_REG14	Interrupt Channel Map Register for 56 to 56+3
0x43c	CH_MAP_REG15	Interrupt Channel Map Register for 60 to 60+3
0x440	CH_MAP_REG16	Interrupt Channel Map Register for 64 to 64+3
0x444	CH_MAP_REG17	Interrupt Channel Map Register for 68 to 68+3
0x448	CH_MAP_REG18	Interrupt Channel Map Register for 72 to 72+3
0x44c	CH_MAP_REG19	Interrupt Channel Map Register for 76 to 76+3
0x450	CH_MAP_REG20	Interrupt Channel Map Register for 80 to 80+3
0x454	CH_MAP_REG21	Interrupt Channel Map Register for 84 to 84+3
0x458	CH_MAP_REG22	Interrupt Channel Map Register for 88 to 88+3
0x45c	CH_MAP_REG23	Interrupt Channel Map Register for 92 to 92+3
0x460	CH_MAP_REG24	Interrupt Channel Map Register for 96 to 96+3
0x464	CH_MAP_REG25	Interrupt Channel Map Register for 100 to 100+3
0x468	CH_MAP_REG26	Interrupt Channel Map Register for 104 to 104+3
0x46c	CH_MAP_REG27	Interrupt Channel Map Register for 108 to 108+3
0x470	CH_MAP_REG28	Interrupt Channel Map Register for 112 to 112+3
0x474	CH_MAP_REG29	Interrupt Channel Map Register for 116 to 116+3
0x478	CH_MAP_REG30	Interrupt Channel Map Register for 120 to 120+3
0x47c	CH_MAP_REG31	Interrupt Channel Map Register for 124 to 124+3
0x480	CH_MAP_REG32	Interrupt Channel Map Register for 128 to 128+3
0x484	CH_MAP_REG33	Interrupt Channel Map Register for 132 to 132+3
0x488	CH_MAP_REG34	Interrupt Channel Map Register for 136 to 136+3
0x48c	CH_MAP_REG35	Interrupt Channel Map Register for 140 to 140+3
0x490	CH_MAP_REG36	Interrupt Channel Map Register for 144 to 144+3
0x494	CH_MAP_REG37	Interrupt Channel Map Register for 148 to 148+3
0x498	CH_MAP_REG38	Interrupt Channel Map Register for 152 to 152+3
0x49c	CH_MAP_REG39	Interrupt Channel Map Register for 156 to 156+3
0x800	HINT_MAP_REG0	Host Interrupt Map Register for 0 to 0+3

Table 7-43 CIC1 Registers (Part 3 of 3)

Address Offset	Register Mnemonic	Register Name
0x804	HINT_MAP_REG1	Host Interrupt Map Register for 4 to 4+3
0x808	HINT_MAP_REG2	Host Interrupt Map Register for 8 to 8+3
0x80c	HINT_MAP_REG3	Host Interrupt Map Register for 12 to 12+3
0x810	HINT_MAP_REG4	Host Interrupt Map Register for 16 to 16+3
0x814	HINT_MAP_REG5	Host Interrupt Map Register for 20 to 20+3
0x818	HINT_MAP_REG6	Host Interrupt Map Register for 24 to 24+3
0x81c	HINT_MAP_REG7	Host Interrupt Map Register for 28 to 28+3
0x820	HINT_MAP_REG8	Host Interrupt Map Register for 32 to 32+3
0x824	HINT_MAP_REG9	Host Interrupt Map Register for 36 to 36+3
0x828	HINT_MAP_REG10	Host Interrupt Map Register for 40 to 40+3
0x82c	HINT_MAP_REG11	Host Interrupt Map Register for 44 to 44+3
0x830	HINT_MAP_REG12	Host Interrupt Map Register for 48 to 48+3
0x834	HINT_MAP_REG13	Host Interrupt Map Register for 52 to 52+3
0x1500	ENABLE_HINT_REG0	Host Int Enable Register 0
0x1504	ENABLE_HINT_REG1	Host Int Enable Register 1
End of Table 7-43		

7.9.2.3 CIC2 Register Map

Table 7-44 CIC2 Registers (Part 1 of 2)

Address Offset	Register Mnemonic	Register Name
0x0	REVISION_REG	Revision Register
0x10	GLOBAL_ENABLE_HINT_REG	Global Host Int Enable Register
0x20	STATUS_SET_INDEX_REG	Status Set Index Register
0x24	STATUS_CLR_INDEX_REG	Status Clear Index Register
0x28	ENABLE_SET_INDEX_REG	Enable Set Index Register
0x2c	ENABLE_CLR_INDEX_REG	Enable Clear Index Register
0x34	HINT_ENABLE_SET_INDEX_REG	Host Int Enable Set Index Register
0x38	HINT_ENABLE_CLR_INDEX_REG	Host Int Enable Clear Index Register
0x200	RAW_STATUS_REG0	Raw Status Register 0
0x204	RAW_STATUS_REG1	Raw Status Register 1
0x280	ENA_STATUS_REG0	Enabled Status Register 0
0x284	ENA_STATUS_REG1	Enabled Status Register 1
0x300	ENABLE_REG0	Enable Register 0
0x304	ENABLE_REG1	Enable Register 1
0x380	ENABLE_CLR_REG0	Enable Clear Register 0
0x384	ENABLE_CLR_REG1	Enable Clear Register 1
0x400	CH_MAP_REG0	Interrupt Channel Map Register for 0 to 0+3
0x404	CH_MAP_REG1	Interrupt Channel Map Register for 4 to 4+3
0x408	CH_MAP_REG2	Interrupt Channel Map Register for 8 to 8+3
0x40c	CH_MAP_REG3	Interrupt Channel Map Register for 12 to 12+3
0x410	CH_MAP_REG4	Interrupt Channel Map Register for 16 to 16+3
0x414	CH_MAP_REG5	Interrupt Channel Map Register for 20 to 20+3
0x418	CH_MAP_REG6	Interrupt Channel Map Register for 24 to 24+3

Table 7-44 CIC2 Registers (Part 2 of 2)

Address Offset	Register Mnemonic	Register Name
0x41c	CH_MAP_REG7	Interrupt Channel Map Register for 28 to 28+3
0x420	CH_MAP_REG8	Interrupt Channel Map Register for 32 to 32+3
0x424	CH_MAP_REG9	Interrupt Channel Map Register for 36 to 36+3
0x428	CH_MAP_REG10	Interrupt Channel Map Register for 40 to 40+3
0x42c	CH_MAP_REG11	Interrupt Channel Map Register for 44 to 44+3
0x430	CH_MAP_REG12	Interrupt Channel Map Register for 48 to 48+3
0x434	CH_MAP_REG13	Interrupt Channel Map Register for 52 to 52+3
0x438	CH_MAP_REG14	Interrupt Channel Map Register for 56 to 56+3
0x43c	CH_MAP_REG15	Interrupt Channel Map Register for 60 to 60+3
0x800	HINT_MAP_REG0	Host Interrupt Map Register for 0 to 0+3
0x804	HINT_MAP_REG1	Host Interrupt Map Register for 4 to 4+3
0x808	HINT_MAP_REG2	Host Interrupt Map Register for 8 to 8+3
0x80c	HINT_MAP_REG3	Host Interrupt Map Register for 12 to 12+3
0x810	HINT_MAP_REG4	Host Interrupt Map Register for 16 to 16+3
0x814	HINT_MAP_REG5	Host Interrupt Map Register for 20 to 20+3
0x818	HINT_MAP_REG6	Host Interrupt Map Register for 24 to 24+3
0x81c	HINT_MAP_REG7	Host Interrupt Map Register for 28 to 28+3
0x820	HINT_MAP_REG8	Host Interrupt Map Register for 32 to 32+3
0x824	HINT_MAP_REG9	Host Interrupt Map Register for 36 to 36+3
0x828	HINT_MAP_REG10	Host Interrupt Map Register for 40 to 40+3
0x1500	ENABLE_HINT_REG0	Host Int Enable Register 0
0x1504	ENABLE_HINT_REG1	Host Int Enable Register 1
End of Table 7-44		

7.9.3 Inter-Processor Register Map

Table 7-45 IPC Generation Registers (IPCGRx) (Part 1 of 2)

Address Start	Address End	Size	Register Name	Description
0x02620200	0x02620203	4B	NMIGR0	NMI Event Generation Register for CorePac0
0x02620204	0x02620207	4B	NMIGR1	NMI Event Generation Register for CorePac1
0x02620208	0x0262020B	4B	NMIGR2	NMI Event Generation Register for CorePac2
0x0262020C	0x0262020F	4B	NMIGR3	NMI Event Generation Register for CorePac3
0x02620210	0x02620213	4B	Reserved	Reserved
0x02620214	0x02620217	4B	Reserved	Reserved
0x02620218	0x0262021B	4B	Reserved	Reserved
0x0262021C	0x0262021F	4B	Reserved	Reserved
0x02620220	0x0262023F	32B	Reserved	Reserved
0x02620240	0x02620243	4B	IPCGR0	IPC Generation Register for CorePac0
0x02620244	0x02620247	4B	IPCGR1	IPC Generation Register for CorePac1
0x02620248	0x0262024B	4B	IPCGR2	IPC Generation Register for CorePac2
0x0262024C	0x0262024F	4B	IPCGR3	IPC Generation Register for CorePac3
0x02620250	0x02620253	4B	Reserved	Reserved
0x02620254	0x02620257	4B	Reserved	Reserved

Table 7-45 IPC Generation Registers (IPCGRx) (Part 2 of 2)

Address Start	Address End	Size	Register Name	Description
0x02620258	0x0262025B	4B	Reserved	Reserved
0x0262025C	0x0262025F	4B	Reserved	Reserved
0x02620260	0x0262027B	28B	Reserved	Reserved
0x0262027C	0x0262027F	4B	IPCGRH	IPC Generation Register for Host
0x02620280	0x02620283	4B	IPCAR0	IPC Acknowledgement Register for CorePac0
0x02620284	0x02620287	4B	IPCAR1	IPC Acknowledgement Register for CorePac1
0x02620288	0x0262028B	4B	IPCAR2	IPC Acknowledgement Register for CorePac2
0x0262028C	0x0262028F	4B	IPCAR3	IPC Acknowledgement Register for CorePac3
0x02620290	0x02620293	4B	Reserved	Reserved
0x02620294	0x02620297	4B	Reserved	Reserved
0x02620298	0x0262029B	4B	Reserved	Reserved
0x0262029C	0x0262029F	4B	Reserved	Reserved
0x026202A0	0x026202BB	28B	Reserved	Reserved
0x026202BC	0x026202BF	4B	IPCARH	IPC Acknowledgement Register for host
End of Table 7-45				

7.9.4 $\overline{\text{NMI}}$ and $\overline{\text{LRESET}}$

The Non-Maskable Interrupts ($\overline{\text{NMI}}$) can be generated by chip-level registers and the $\overline{\text{LRESET}}$ can be generated by software writing into LPSC registers. $\overline{\text{LRESET}}$ and $\overline{\text{NMI}}$ can also be asserted by device pins or watchdog timers. One $\overline{\text{NMI}}$ pin and one $\overline{\text{LRESET}}$ pin are shared by all four CorePacs on the device. The CORESEL[2:0] pins can be configured to select between the four CorePacs available as shown in Table 7-46.

Table 7-46 $\overline{\text{LRESET}}$ and $\overline{\text{NMI}}$ Decoding

CORESEL[2:0] Pin Input	$\overline{\text{LRESET}}$ Pin Input	$\overline{\text{NMI}}$ Pin Input	$\overline{\text{LRESETNMIEN}}$ Pin Input	Reset Mux Block Output
XXX	X	X	1	No local reset or $\overline{\text{NMI}}$ assertion
000	0	X	0	Assert local reset to CorePac0
001	0	X	0	Assert local reset to CorePac1
010	0	X	0	Assert local reset to CorePac2
011	0	X	0	Assert local reset to CorePac3
1xx	0	X	0	Assert local reset to all CorePacs
000	1	1	0	De-assert local reset & $\overline{\text{NMI}}$ to CorePac0
001	1	1	0	De-assert local reset & $\overline{\text{NMI}}$ to CorePac1
010	1	1	0	De-assert local reset & $\overline{\text{NMI}}$ to CorePac2
011	1	1	0	De-assert local reset & $\overline{\text{NMI}}$ to CorePac3
1xx	1	1	0	De-assert local reset & $\overline{\text{NMI}}$ to all CorePacs
000	1	0	0	Assert $\overline{\text{NMI}}$ to CorePac0
001	1	0	0	Assert $\overline{\text{NMI}}$ to CorePac1
010	1	0	0	Assert $\overline{\text{NMI}}$ to CorePac2
011	1	0	0	Assert $\overline{\text{NMI}}$ to CorePac3
1xx	1	0	0	Assert $\overline{\text{NMI}}$ to all CorePacs
End of Table 7-46				

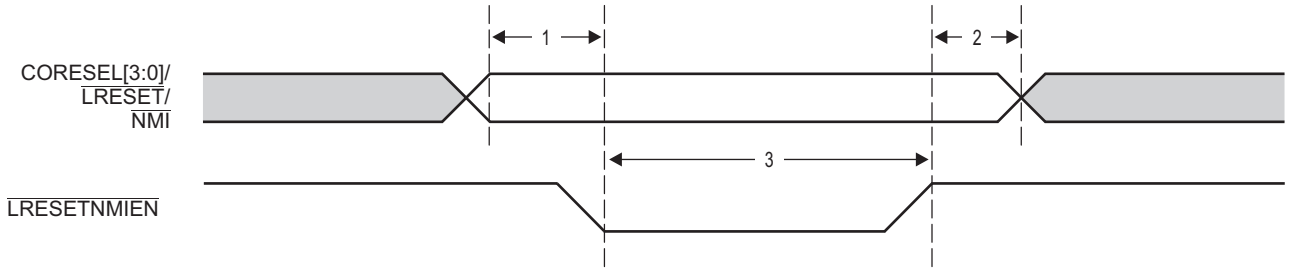
7.9.5 External Interrupts Electrical Data/Timing

Table 7-47 $\overline{\text{NMI}}$ and $\overline{\text{LRESET}}$ Timing Requirements (1)
(see Figure 7-30)

No.			Min	Max	Unit
1	tsu($\overline{\text{LRESET}}$-$\overline{\text{LRESETNMIENL}}$)	Setup time - $\overline{\text{LRESET}}$ valid before $\overline{\text{LRESETNMIEN}}$ low	12*P		ns
1	tsu($\overline{\text{NMI}}$-$\overline{\text{LRESETNMIENL}}$)	Setup time - $\overline{\text{NMI}}$ valid before $\overline{\text{LRESETNMIEN}}$ low	12*P		ns
1	tsu(CORESELn-$\overline{\text{LRESETNMIENL}}$)	Setup time - CORESEL[2:0] valid before $\overline{\text{LRESETNMIEN}}$ low	12*P		ns
2	th($\overline{\text{LRESETNMIENL}}$-$\overline{\text{LRESET}}$)	Hold time - $\overline{\text{LRESET}}$ valid after $\overline{\text{LRESETNMIEN}}$ high	12*P		ns
2	th($\overline{\text{LRESETNMIENL}}$-$\overline{\text{NMI}}$)	Hold time - $\overline{\text{NMI}}$ valid after $\overline{\text{LRESETNMIEN}}$ high	12*P		ns
2	th($\overline{\text{LRESETNMIENL}}$-CORESELn)	Hold time - CORESEL[2:0] valid after $\overline{\text{LRESETNMIEN}}$ high	12*P		ns
3	tw($\overline{\text{LRESETNMIEN}}$)	Pulsewidth - $\overline{\text{LRESETNMIEN}}$ low width	12*P		ns
End of Table 7-47					

1 P = 1/SYSCLK1 clock frequency in ns.

Figure 7-30 $\overline{\text{NMI}}$ and $\overline{\text{LRESET}}$ Timing



7.10 Memory Protection Unit (MPU)

The C6670 supports six MPUs:

- One MPU is used to protect main CORE/3 CFG TeraNet (CFG space of all slave devices on the TeraNet is protected by the MPU).
- Two MPUs are used for packet DMA (one for DATA PORT and another is for CFG PORT).
- One MPU is used for Semaphore.
- One MPU is used for the RAC.
- One MPU is used to protect the main CFG TeraNet of the TE_SCR_3M.

This section contains MPU register map and details of device-specific MPU registers only. For MPU features and details of generic MPU registers, see the *Memory Protection Unit (MPU) for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

The following tables show the configuration of each MPU and the memory regions protected by each MPU.

Table 7-48 MPU Default Configuration

Setting	MPU0 Main CFG SCR	MPU1 (QM_SS DATA PORT)	MPU2 (QM_SS CFG PORT)	MPU3 Semaphore	MPU4 RAC	MPU5 TE_SCR_3M
Default permission	Assume allowed	Assume allowed	Assume allowed	Assume allowed	Assume allowed	Assume allowed
Number of allowed IDs supported	16	16	16	16	16	16
Number of programmable ranges supported	16	5	16	1	2	3
Compare width	1KB granularity	1KB granularity	1KB granularity	1KB granularity	1KB granularity	1KB granularity
End of Table 7-48						

Table 7-49 MPU Memory Regions

	Memory Protection	Start Address	End Address
MPU0	Main CFG SCR	0x01D00000	0x026203FF
MPU1	QM_SS DATA PORT	0x34000000	0x340BFFFF
MPU2	QM_SS CFG PORT	0x02A00000	0x02ABFFFF
MPU3	Semaphore	0x02640000	0x026407FF
MPU4	RAC	0x01F80000	0x0215FFFF
MPU5	TE_SCR_3M	0x35000000	0x350003FF
End of Table 7-49			

Table 7-50 shows the unique Master ID assigned to each CorePac and peripherals on the device.

Table 7-50 Master ID Settings (Part 1 of 3)

Master ID	C6670
0	CorePac0
1	CorePac1
2	CorePac2
3	CorePac3
4	Reserved
5	Reserved
6	Reserved
7	Reserved

Table 7-50 Master ID Settings (Part 2 of 3)

Master ID	C6670
8	CorePac0 CFG
9	CorePac1 CFG
10	CorePac2 CFG
11	CorePac3 CFG
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	EDMA0_TC0 read
17	EDMA0_TC0 write
18	EDMA0_TC1 read
19	EDMA0_TC1 write
20	EDMA1_TC0 read
21	EDMA1_TC0 write
22	EDMA1_TC1 read
23	EDMA1_TC1write
24	EDMA1_TC2 read
25	EDMA1_TC2 write
26	EDMA1_TC3 read
27	EDMA1_TC3 write
28	EDMA2_TC0 read
29	EDMA2_TC0 write
30	EDMA2_TC1 read
31	EDMA2_TC1 write
32	EDMA2_TC2 read
33	EDMA2_TC2 write
34	EDMA2_TC3 read
35	EDMA2_TC3 write
36 to 37	Reserved
38 to 39	SRIO PKTDMA
40	FFTC_A
41	Reserved
42	FFTC_B
43	Reserved
44	RAC_B_BE0
45	RAC_B_BE1
46	RAC_A_BE0
47	RAC_A_BE1
48	DebugSS
49	EDMA3CC0
50	EDMA3CC1
51	EDMA3CC2
52	MSMC (1)
53	PCIe

Table 7-50 Master ID Settings (Part 3 of 3)

Master ID	C6670
54	SRIO_M
55	HyperLink
56 to 59	Queue Manager
60 to 63	Reserved
64 to 71	AIF2
72 to 85	Reserved
86	Reserved
87	Reserved
88 to 91	Queue Manager packet DMA
92 to 93	Packet Coprocessor
94	TAC
95	Reserved
96 to 127	TE_SS
128	Tracer_L2_0 (2)
129	Tracer_L2_1
130	Tracer_L2_2
131	Tracer_L2_3
132	Reserved
133	Reserved
134	Reserved
135	Reserved
136	Tracer_MSMC0
137	Tracer_MSMC1
138	Tracer_MSMC2
139	Tracer_MSMC3
140	Tracer_DDR
141	Tracer_SM
142	Tracer_QM_P
143	Tracer_QM_M
144	Tracer_CFG
145	Tracer_RAC
146	Tracer_RAC_CFG
147	Tracer_TAC
End of Table 7-50	

1 The master ID for MSMC is for the transactions initiated by MSMC internally and sent to the DDR.
2 All Traces are set to the same master ID and bit 7 of the master ID needs to be 1.

Note—Some of the PKTDMA based-peripherals require multiple master IDs. Queue Manager Packet DMA is assigned with 88, 89, 90, 91, but only 88-89 are actually used. For Queue Manager port, 56, 57, 58, 59 are assigned while only 1 (56) is actually used. For AIF2, 64, 65, 66, 67, 68, 69, 70, 71 are assigned while only 4 (64-67) are actually used. There are two master ID values are assigned for the Queue Manager_second master port, one master ID for external linking RAM and the other one for the PDSP/MCDM accesses.

Table 7-51 shows the privilege ID of each CorePac and every mastering peripheral. Table 7-51 also shows the privilege level (supervisor vs. user), security level (secure vs. non-secure), and access type (instruction read vs. data/DMA read or write) of each master on the device. In some cases, a particular setting depends on software being executed at the time of the access or the configuration of the master peripheral.

Table 7-51 Privilege ID Settings

Privilege ID	Master	Privilege Level	Security Level	Access Type
0	CorePac0	SW dependant, driven by MSMC	SW dependant	DMA
1	CorePac1	SW dependant, driven by MSMC	SW dependant	DMA
2	CorePac2	SW dependant, driven by MSMC	SW dependant	DMA
3	CorePac3	SW dependant, driven by MSMC	SW dependant	DMA
4	AIF	User	Non-secure	DMA
5	TAC	User	Non-secure	DMA
6	RAC	User	Non-secure	DMA
7	FFTC	User	Non-secure	DMA
8	QM_SS Second	User	Non-secure	DMA
9	SRIO Packet DMA/SRIO_M	User/driven by SRIO block, user mode and supervisor mode is determined by per transaction basis. Only the transaction with source ID matching the value in SupervisorID register is granted supervisor mode.	Non-secure	DMA
10	QM_SS Packet DMA/NETCP Packet DMA	User	Non-secure	DMA
11	PCIe	Supervisor	Non-secure	DMA
12	DebugSS	Driven by Debug_SS	Driven by debug_SS	DMA
13	HyperLink	Supervisor	Non-secure	DMA
14	HyperLink	Supervisor	Non-secure	DMA
15	TE_SCR_3M	User	Non-secure	DMA
End of Table 7-51				

7.10.1 MPU Registers

This section includes the offsets for MPU registers and definitions for device-specific MPU registers.

7.10.1.1 MPU Register Map

Table 7-52 MPU0 Registers (Part 1 of 3)

Offset	Name	Description
0h	REVID	Revision ID
4h	CONFIG	Configuration
10h	IRAWSTAT	Interrupt raw status/set
14h	IENSTAT	Interrupt enable status/clear
18h	IENSET	Interrupt enable
1Ch	IENCLR	Interrupt enable clear
20h	EOI	End of interrupt
200h	PROG0_MPSAR	Programmable range 0, start address
204h	PROG0_MPEAR	Programmable range 0, end address
208h	PROG0_MPPA	Programmable range 0, memory page protection attributes
210h	PROG1_MPSAR	Programmable range 1, start address
214h	PROG1_MPEAR	Programmable range 1, end address

Table 7-52 MPU0 Registers (Part 2 of 3)

Offset	Name	Description
218h	PROG1_MPPA	Programmable range 1, memory page protection attributes
220h	PROG2_MPSAR	Programmable range 2, start address
224h	PROG2_MPEAR	Programmable range 2, end address
228h	PROG2_MPPA	Programmable range 2, memory page protection attributes
230h	PROG3_MPSAR	Programmable range 3, start address
234h	PROG3_MPEAR	Programmable range 3, end address
238h	PROG3_MPPA	Programmable range 3, memory page protection attributes
240h	PROG4_MPSAR	Programmable range 4, start address
244h	PROG4_MPEAR	Programmable range 4, end address
248h	PROG4_MPPA	Programmable range 4, memory page protection attributes
250h	PROG5_MPSAR	Programmable range 5, start address
254h	PROG5_MPEAR	Programmable range 5, end address
258h	PROG5_MPPA	Programmable range 5, memory page protection attributes
260h	PROG6_MPSAR	Programmable range 6, start address
264h	PROG6_MPEAR	Programmable range 6, end address
268h	PROG6_MPPA	Programmable range 6, memory page protection attributes
270h	PROG7_MPSAR	Programmable range 7, start address
274h	PROG7_MPEAR	Programmable range 7, end address
278h	PROG7_MPPA	Programmable range 7, memory page protection attributes
280h	PROG8_MPSAR	Programmable range 8, start address
284h	PROG8_MPEAR	Programmable range 8, end address
288h	PROG8_MPPA	Programmable range 8, memory page protection attributes
290h	PROG9_MPSAR	Programmable range 9, start address
294h	PROG9_MPEAR	Programmable range 9, end address
298h	PROG9_MPPA	Programmable range 9, memory page protection attributes
2A0h	PROG10_MPSAR	Programmable range 10, start address
2A4h	PROG10_MPEAR	Programmable range 10, end address
2A8h	PROG10_MPPA	Programmable range 10, memory page protection attributes
2B0h	PROG11_MPSAR	Programmable range 11, start address
2B4h	PROG11_MPEAR	Programmable range 11, end address
2B8h	PROG11_MPPA	Programmable range 11, memory page protection attributes
2C0h	PROG12_MPSAR	Programmable range 12, start address
2C4h	PROG12_MPEAR	Programmable range 12, end address
2C8h	PROG12_MPPA	Programmable range 12, memory page protection attributes
2D0h	PROG13_MPSAR	Programmable range 13, start address
2D4h	PROG13_MPEAR	Programmable range 13, end address
2Dh	PROG13_MPPA	Programmable range 13, memory page protection attributes
2E0h	PROG14_MPSAR	Programmable range 14, start address
2E4h	PROG14_MPEAR	Programmable range 14, end address
2E8h	PROG14_MPPA	Programmable range 14, memory page protection attributes
2F0h	PROG15_MPSAR	Programmable range 15, start address
2F4h	PROG15_MPEAR	Programmable range 15, end address
2F8h	PROG15_MPPA	Programmable range 15, memory page protection attributes
300h	FLTADDRR	Fault address

Table 7-52 MPU0 Registers (Part 3 of 3)

Offset	Name	Description
304h	FLTSTAT	Fault status
308h	FLTCLR	Fault clear
End of Table 7-52		

Table 7-53 MPU1 Registers

Offset	Name	Description
0h	REVID	Revision ID
4h	CONFIG	Configuration
10h	IRAWSTAT	Interrupt raw status/set
14h	IENSTAT	Interrupt enable status/clear
18h	IENSET	Interrupt enable
1Ch	IENCLR	Interrupt enable clear
20h	EOI	End of interrupt
200h	PROG0_MPSAR	Programmable range 0, start address
204h	PROG0_MPEAR	Programmable range 0, end address
208h	PROG0_MPPA	Programmable range 0, memory page protection attributes
210h	PROG1_MPSAR	Programmable range 1, start address
214h	PROG1_MPEAR	Programmable range 1, end address
218h	PROG1_MPPA	Programmable range 1, memory page protection attributes
220h	PROG2_MPSAR	Programmable range 2, start address
224h	PROG2_MPEAR	Programmable range 2, end address
228h	PROG2_MPPA	Programmable range 2, memory page protection attributes
230h	PROG3_MPSAR	Programmable range 3, start address
234h	PROG3_MPEAR	Programmable range 3, end address
238h	PROG3_MPPA	Programmable range 3, memory page protection attributes
300h	FLTADDRR	Fault address
304h	FLTSTAT	Fault status
308h	FLTCLR	Fault clear
End of Table 7-53		

Table 7-54 MPU2 Registers (Part 1 of 3)

Offset	Name	Description
0h	REVID	Revision ID
4h	CONFIG	Configuration
10h	IRAWSTAT	Interrupt raw status/set
14h	IENSTAT	Interrupt enable status/clear
18h	IENSET	Interrupt enable
1Ch	IENCLR	Interrupt enable clear
20h	EOI	End of interrupt
200h	PROG0_MPSAR	Programmable range 0, start address
204h	PROG0_MPEAR	Programmable range 0, end address
208h	PROG0_MPPA	Programmable range 0, memory page protection attributes
210h	PROG1_MPSAR	Programmable range 1, start address

Table 7-54 MPU2 Registers (Part 2 of 3)

Offset	Name	Description
214h	PROG1_MPEAR	Programmable range 1, end address
218h	PROG1_MPPA	Programmable range 1, memory page protection attributes
220h	PROG2_MPSAR	Programmable range 2, start address
224h	PROG2_MPEAR	Programmable range 2, end address
228h	PROG2_MPPA	Programmable range 2, memory page protection attributes
230h	PROG3_MPSAR	Programmable range 3, start address
234h	PROG3_MPEAR	Programmable range 3, end address
238h	PROG3_MPPA	Programmable range 3, memory page protection attributes
240h	PROG4_MPSAR	Programmable range 4, start address
244h	PROG4_MPEAR	Programmable range 4, end address
248h	PROG4_MPPA	Programmable range 4, memory page protection attributes
250h	PROG5_MPSAR	Programmable range 5, start address
254h	PROG5_MPEAR	Programmable range 5, end address
258h	PROG5_MPPA	Programmable range 5, memory page protection attributes
260h	PROG6_MPSAR	Programmable range 6, start address
264h	PROG6_MPEAR	Programmable range 6, end address
268h	PROG6_MPPA	Programmable range 6, memory page protection attributes
270h	PROG7_MPSAR	Programmable range 7, start address
274h	PROG7_MPEAR	Programmable range 7, end address
278h	PROG7_MPPA	Programmable range 7, memory page protection attributes
280h	PROG8_MPSAR	Programmable range 8, start address
284h	PROG8_MPEAR	Programmable range 8, end address
288h	PROG8_MPPA	Programmable range 8, memory page protection attributes
290h	PROG9_MPSAR	Programmable range 9, start address
294h	PROG9_MPEAR	Programmable range 9, end address
298h	PROG9_MPPA	Programmable range 9, memory page protection attributes
2A0h	PROG10_MPSAR	Programmable range 10, start address
2A4h	PROG10_MPEAR	Programmable range 10, end address
2A8h	PROG10_MPPA	Programmable range 10, memory page protection attributes
2B0h	PROG11_MPSAR	Programmable range 11, start address
2B4h	PROG11_MPEAR	Programmable range 11, end address
2B8h	PROG11_MPPA	Programmable range 11, memory page protection attributes
2C0h	PROG12_MPSAR	Programmable range 12, start address
2C4h	PROG12_MPEAR	Programmable range 12, end address
2C8h	PROG12_MPPA	Programmable range 12, memory page protection attributes
2D0h	PROG13_MPSAR	Programmable range 13, start address
2D4h	PROG13_MPEAR	Programmable range 13, end address
2Dh	PROG13_MPPA	Programmable range 13, memory page protection attributes
2E0h	PROG14_MPSAR	Programmable range 14, start address
2E4h	PROG14_MPEAR	Programmable range 14, end address
2E8h	PROG14_MPPA	Programmable range 14, memory page protection attributes
2F0h	PROG15_MPSAR	Programmable range 15, start address
2F4h	PROG15_MPEAR	Programmable range 15, end address
2F8h	PROG15_MPPA	Programmable range 15, memory page protection attributes

Table 7-54 MPU2 Registers (Part 3 of 3)

Offset	Name	Description
300h	FLTADDRR	Fault address
304h	FLTSTAT	Fault status
308h	FLTCLR	Fault clear
End of Table 7-54		

Table 7-55 MPU3 Registers

Offset	Name	Description
0h	REVID	Revision ID
4h	CONFIG	Configuration
10h	IRAWSTAT	Interrupt raw status/set
14h	IENSTAT	Interrupt enable status/clear
18h	IENSET	Interrupt enable
1Ch	IENCLR	Interrupt enable clear
20h	EOI	End of interrupt
200h	PROG0_MPSAR	Programmable range 0, start address
204h	PROG0_MPEAR	Programmable range 0, end address
208h	PROG0_MPPA	Programmable range 0, memory page protection attributes
300h	FLTADDRR	Fault address
304h	FLTSTAT	Fault status
308h	FLTCLR	Fault clear
End of Table 7-55		

Table 7-56 MPU4 Registers

Offset	Name	Description
0h	REVID	Revision ID
4h	CONFIG	Configuration
10h	IRAWSTAT	Interrupt raw status/set
14h	IENSTAT	Interrupt enable status/clear
18h	IENSET	Interrupt enable
1Ch	IENCLR	Interrupt enable clear
20h	EOI	End of interrupt
200h	PROG0_MPSAR	Programmable range 0, start address
204h	PROG0_MPEAR	Programmable range 0, end address
208h	PROG0_MPPA	Programmable range 0, memory page protection attributes
210h	PROG1_MPSAR	Programmable range 1, start address
214h	PROG1_MPEAR	Programmable range 1, end address
218h	PROG1_MPPA	Programmable range 1, memory page protection attributes
300h	FLTADDRR	Fault address
304h	FLTSTAT	Fault status
308h	FLTCLR	Fault clear
End of Table 7-56		

Table 7-57 MPU5 Registers

Offset	Name	Description
0h	REVID	Revision ID
4h	CONFIG	Configuration
10h	IRAWSTAT	Interrupt raw status/set
14h	IENSTAT	Interrupt enable status/clear
18h	IENSET	Interrupt enable
1Ch	IENCLR	Interrupt enable clear
20h	EOI	End of interrupt
200h	PROG0_MPSAR	Programmable range 0, start address
204h	PROG0_MPEAR	Programmable range 0, end address
208h	PROG0_MPPA	Programmable range 0, memory page protection attributes
210h	PROG1_MPSAR	Programmable range 1, start address
214h	PROG1_MPEAR	Programmable range 1, end address
218h	PROG1_MPPA	Programmable range 1, memory page protection attributes
220h	PROG2_MPSAR	Programmable range 2, start address
224h	PROG2_MPEAR	Programmable range 2, end address
228h	PROG2_MPPA	Programmable range 2, memory page protection attributes
300h	FLTADDRR	Fault address
304h	FLTSTAT	Fault status
308h	FLTCLR	Fault clear
End of Table 7-57		

7.10.1.2 Device-Specific MPU Registers

7.10.1.2.1 Configuration Register (CONFIG)

The configuration register (CONFIG) contains the configuration value of the MPU.

Figure 7-31 Configuration Register (CONFIG)

		31–24	23–20	19–16	15–12	11–1	0
		ADDR_WIDTH	NUM_FIXED	NUM_PROG	NUM_AIDS	Reserved	ASSUME_ALLOWED
Reset Values	MPU0	R-0	R-0	R-16	R-16	R-0	R-1
	MPU1	R-0	R-0	R-5	R-16	R-0	R-1
	MPU2	R-0	R-0	R-16	R-16	R-0	R-1
	MPU3	R-0	R-0	R-1	R-16	R-0	R-1
	MPU4	R-0	R-0	R-2	R-16	R-0	R-1
	MPU5	R-0	R-0	R-3	R-16	R-0	R-1

Legend: R = Read only; -*n* = value after reset

Table 7-58 Configuration Register Field Descriptions

Bits	Field	Description
31 – 24	ADDR_WIDTH	Address alignment for range checking 0 = 1KB alignment 6 = 64KB alignment
23 – 20	NUM_FIXED	Number of fixed address ranges
19 – 16	NUM_PROG	Number of programmable address ranges
15 – 12	NUM_AIDS	Number of supported AIDs
11 – 1	Reserved	Reserved. Always read as 0.
0	ASSUME_ALLOWED	Assume allowed bit. When an address is not covered by any MPU protection range, this bit determines whether the transfer is assumed to be allowed or not. 0 = Assume disallowed 1 = Assume allowed

7.10.2 MPU Programmable Range Registers

7.10.2.1 Programmable Range *n* Start Address Register (PROG*n*_MPSAR)

The Programmable Address Start Register holds the start address for the range. This register is writeable by a supervisor entity only. If NS = 0 (non-secure mode) in the associated MPPA register, then the register is also writeable only by a secure entity.

The start address must be aligned on a page boundary. The size of the page is 1K byte. The size of the page determines the width of the address field in MPSAR and MPEAR.

Figure 7-32 Programmable Range *n* Start Address Register (PROG*n*_MPSAR)

31 – 10	9 – 0
START_ADDR	Reserved
R/W	R

Legend: R = Read only; R/W = Read/Write

Table 7-59 Programmable Range *n* Start Address Register Field Descriptions

Bit	Field	Description
31 – 10	START_ADDR	Start address for range n
9 – 0	Reserved	Reserved. Always read as 0.
End of Table 7-59		

Table 7-60 Programmable Range *n* Start Address Register (PROG*n*_MPSAR) Reset Values

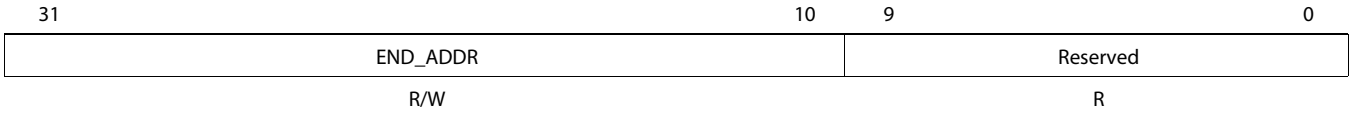
Register	MPU0	MPU1	MPU2	MPU3	MPU4	MPU5
PROG0_MPSAR	0x01D0_0000	0x3400_0000	0x02A0_0000	0x0264_0000	0x0210_0000	0x3502_0000
PROG1_MPSAR	0x01F0_0000	0x3402_0000	0x02A2_0000	N/A	0x01F8_0000	0x3504_0000
PROG2_MPSAR	0x0200_0000	0x3406_0000	0x02A4_0000	N/A	N/A	0x3520_0000
PROG3_MPSAR	0x0218_0000	0x3406_8000	0x02A6_0000	N/A	N/A	
PROG4_MPSAR	0x021C_0000	0x340B_8000	0x02A6_8000	N/A	N/A	
PROG5_MPSAR	0x021F_0000	N/A	0x02A6_9000	N/A	N/A	
PROG6_MPSAR	0x0220_0000	N/A	0x02A6_A000	N/A	N/A	
PROG7_MPSAR	0x0231_0000	N/A	0x02A6_B000	N/A	N/A	
PROG8_MPSAR	0x0232_0000	N/A	0x02A6_C000	N/A	N/A	
PROG9_MPSAR	0x0233_0000	N/A	0x02A6_E000	N/A	N/A	
PROG10_MPSAR	0x0235_0000	N/A	0x02A8_0000	N/A	N/A	
PROG11_MPSAR	0x0240_0000	N/A	0x02A9_0000	N/A	N/A	
PROG12_MPSAR	0x0250_0000	N/A	0x02AA_0000	N/A	N/A	
PROG13_MPSAR	0x0253_0000	N/A	0x02AA_8000	N/A	N/A	
PROG14_MPSAR	0x0260_0000	N/A	0x02AB_0000	N/A	N/A	
PROG15_MPSAR	0x0262_0000	N/A	0x02AB_8000	N/A	N/A	
End of Table 7-60						

7.10.2.2 Programmable Range *n* - End Address Register (PROG*n*_MPEAR)

The programmable address end register holds the end address for the range. This register is writeable by a supervisor entity only. If NS = 0 (non-secure mode) in the associated MPPA register then the register is also writeable only by a secure entity.

The end address must be aligned on a page boundary. The size of the page depends on the MPU number. The page size for MPU1 is 1K byte and for MPU2 it is 64K bytes. The size of the page determines the width of the address field in MPSAR and MPEAR

Figure 7-33 Programmable Range *n* End Address Register (PROG*n*_MPEAR)



Legend: R = Read only; R/W = Read/Write

Table 7-61 Programmable Range *n* End Address Register Field Descriptions

Bit	Field	Description
31 – 10	END_ADDR	End address for range n
9 – 0	Reserved	Reserved. Always read as 3FFh.
End of Table 7-61		

Table 7-62 Programmable Range *n* End Address Register (PROG*n*_MPEAR) Reset Values

Register	MPU0	MPU1	MPU2	MPU3	MPU4	MPU5
PROG0_MPEAR	0x01D8_03FF	0x3401_FFFF	0x02A1_FFFF	0x0264_07FF	0x0215_FFFF	0x3502_03FF
PROG1_MPEAR	0x01F7_FFFF	0x3405_FFFF	0x02A3_FFFF	N/A	0x01FD_FFFF	0x3504_07FF
PROG2_MPEAR	0x0209_FFFF	0x3406_7FFF	0x02A5_FFFF	N/A	N/A	0x3521_FFFF
PROG3_MPEAR	0x021A_FFFF	0x340B_7FFF	0x02A6_7FFF	N/A	N/A	
PROG4_MPEAR	0x021E_0FFF	0x340B_FFFF	0x02A6_8FFF	N/A	N/A	
PROG5_MPEAR	0x021F_7FFF	N/A	0x02A6_9FFF	N/A	N/A	
PROG6_MPEAR	0x022F_03FF	N/A	0x02A6_AFFF	N/A	N/A	
PROG7_MPEAR	0x0231_03FF	N/A	0x02A6_BFFF	N/A	N/A	
PROG8_MPEAR	0x0232_03FF	N/A	0x02A6_DFFF	N/A	N/A	
PROG9_MPEAR	0x0233_03FF	N/A	0x02A6_FFFF	N/A	N/A	
PROG10_MPEAR	0x0235_0FFF	N/A	0x02A8_FFFF	N/A	N/A	
PROG11_MPEAR	0x024B_3FFF	N/A	0x02A9_FFFF	N/A	N/A	
PROG12_MPEAR	0x0252_03FF	N/A	0x02AA_7FFF	N/A	N/A	
PROG13_MPEAR	0x0254_03FF	N/A	0x02AA_FFFF	N/A	N/A	
PROG14_MPEAR	0x0260_FFFF	N/A	0x02AB_7FFF	N/A	N/A	
PROG15_MPEAR	0x0262_07FF	N/A	0x02AB_FFFF	N/A	N/A	
End of Table 7-62						

7.10.2.3 Programmable Range *n* Memory Protection Page Attribute Register (PROG*n*_MPPA)

The programmable address memory protection page attribute register holds the permissions for the region. This register is writeable only by a non-debug supervisor entity. If NS = 0 (secure mode) then the register is also writeable only by a non-debug secure entity. The NS bit is writeable only by a non-debug secure entity. For debug accesses, the register is writeable only when NS = 1 or EMU = 1.

Figure 7-34 Programmable Range *n* Memory Protection Page Attribute Register (PROG*n*_MPPA)

31	26	25	24	23	22	21	20	19	18	17	16	15
Reserved		AID15	AID14	AID13	AID12	AID11	AID10	AID9	AID8	AID7	AID6	AID5
R		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AID4	AID3	AID2	AID1	AID0	AIDX	Reserved	NS	EMU	SR	SW	SX	UR	UW	UX
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Legend: R = Read only; R/W = Read/Write

Table 7-63 Programmable Range *n* Memory Protection Page Attribute Register Field Descriptions (Part 1 of 2)

Bits	Name	Description
31 – 26	Reserved	Reserved. Always read as 0.
25	AID15	Controls access from ID = 15 0 = Access denied 1 = Access granted
24	AID14	Controls access from ID = 14 0 = Access denied 1 = Access granted
23	AID13	Controls access from ID = 13 0 = Access denied 1 = Access granted
22	AID12	Controls access from ID = 12 0 = Access denied 1 = Access granted
21	AID11	Controls access from ID = 11 0 = Access denied 1 = Access granted
20	AID10	Controls access from ID = 10 0 = Access denied 1 = Access granted
19	AID9	Controls access from ID = 9 0 = Access denied 1 = Access granted
18	AID8	Controls access from ID = 8 0 = Access denied 1 = Access granted
17	AID7	Controls access from ID = 7 0 = Access denied 1 = Access granted
16	AID6	Controls access from ID = 6 0 = Access denied 1 = Access granted
15	AID5	Controls access from ID = 5 0 = Access denied 1 = Access granted

Table 7-63 Programmable Range *n* Memory Protection Page Attribute Register Field Descriptions (Part 2 of 2)

Bits	Name	Description
14	AID4	Controls access from ID = 4 0 = Access denied 1 = Access granted
13	AID3	Controls access from ID = 3 0 = Access denied 1 = Access granted
12	AID2	Controls access from ID = 2 0 = Access denied 1 = Access granted
11	AID1	Controls access from ID = 1 0 = Access denied 1 = Access granted
10	AID0	Controls access from ID = 0 0 = Access denied 1 = Access granted
9	AIDX	Controls access from ID > 15 0 = Access denied 1 = Access granted
8	Reserved	Reserved. Always reads as 0.
7	NS	Non-secure access permission 0 = Only secure access allowed 1 = Non-secure access allowed
6	EMU	Emulation (debug) access permission. This bit is ignored if NS = 1 0 = Debug access not allowed 1 = Debug access allowed
5	SR	Supervisor Read permission 0 = Access not allowed 1 = Access allowed
4	SW	Supervisor Write permission 0 = Access not allowed 1 = Access allowed
3	SX	Supervisor Execute permission 0 = Access not allowed 1 = Access allowed
2	UR	User Read permission 0 = Access not allowed 1 = Access allowed
1	UW	User Write permission 0 = Access not allowed 1 = Access allowed
0	UX	User Execute permission 0 = Access not allowed 1 = Access allowed
End of Table 7-631		

Table 7-64 Programmable Range *n* Memory Protection Page Attribute Register (PROG*n*_MPPA) Reset Values

Register	MPU0	MPU1	MPU2	MPU3	MPU4	MPU5
Register 0	0X03FF_FCB6	0X03FF_FC80	0x03FF_FCA4	0X0003_FCB6	0X0003_FCB6	0X03FF_FCB6
Register 1	0X03FF_FCB6	0X0003_FCB6	0X0003_FCB6	N/A	0X0003_FCB6	0X03FF_FCB6
Register 2	0X03FF_FCB6	0X0003_FCB4	0X0003_FCB6	N/A	N/A	0X03FF_FCB6
Register 3	0X03FF_FCB6	0X0003_FC80	0X0003_FCB4	N/A	N/A	N/A
Register 4	0X03FF_FCB6	0X0003_FCB6	0X0003_FCB4	N/A	N/A	N/A
Register 5	0X03FF_FCB6	N/A	0X0003_FCB4	N/A	N/A	N/A
Register 6	0X03FF_FCB6	N/A	0X0003_FCB4	N/A	N/A	N/A
Register 7	0X03FF_FCB4	N/A	0X0003_FCB4	N/A	N/A	N/A
Register 8	0X03FF_FCB4	N/A	0X0003_FCB4	N/A	N/A	N/A
Register 9	0X03FF_FCB4	N/A	0X0003_FCB4	N/A	N/A	N/A
Register 10	0X03FF_FCB4	N/A	0X0003_FCA4	N/A	N/A	N/A
Register 11	0X03FF_FCB6	N/A	0X0003_FCB4	N/A	N/A	N/A
Register 12	0X03FF_FCB4	N/A	0X0003_FCB4	N/A	N/A	N/A
Register 13	0X03FF_FCB6	N/A	0X0003_FCB4	N/A	N/A	N/A
Register 14	0X03FF_FCB4	N/A	0X0003_FCB4	N/A	N/A	N/A
Register 15	0X03FF_FCB4	N/A	0X0003_FCB6	N/A	N/A	N/A
End of Table 7-64						

7.11 DDR3 Memory Controller

The 64-bit DDR3 Memory Controller bus of the TMS320C6670 is used to interface to JEDEC standard-compliant DDR3 SDRAM devices. The DDR3 external bus interfaces only to DDR3 SDRAM devices; it does not share the bus with any other types of peripherals.

7.11.1 DDR3 Memory Controller Device-Specific Information

The TMS320C6670 includes one 64-bit wide, 1.5-V DDR3 SDRAM EMIF interface. The DDR3 interface can operate at 800 mega transfers per second (MTS), 1033 MTS, 1333 MTS, and 1600 MTS.

Due to the complicated nature of the interface, a limited number of topologies will be supported to provide a 16-bit, 32-bit, or 64-bit interface.

The DDR3 electrical requirements are fully specified in the DDR Jedec Specification JESD79-3C. Standard DDR3 SDRAMs are available in 8-bit and 16-bit versions, allowing for the following bank topologies to be supported by the interface:

- 72-bit: Five 16-bit SDRAMs (including 8 bits of ECC)
- 72-bit: Nine 8-bit SDRAMs (including 8 bits of ECC)
- 36-bit: Three 16-bit SDRAMs (including 4 bits of ECC)
- 36-bit: Five 8-bit SDRAMs (including 4 bits of ECC)
- 64-bit: Four 16-bit SDRAMs
- 64-bit: Eight 8-bit SDRAMs
- 32-bit: Two 16-bit SDRAMs
- 32-bit: Four 8-bit SDRAMs
- 16-bit: One 16-bit SDRAM
- 16-bit: Two 8-bit SDRAMs

The approach to specifying interface timing for the DDR3 memory bus is different than on other interfaces such as I^2C or SPI. For these other interfaces, the device timing was specified in terms of data manual specifications and I/O buffer information specification (IBIS) models. For the DDR3 memory bus, the approach is to specify compatible DDR3 devices and provide the printed circuit board (PCB) solution and guidelines directly to the user.

A race condition may exist when certain masters write data to the DDR3 memory controller. For example, if master A passes a software message via a buffer in external memory and does not wait for an indication that the write completes, before signaling to master B that the message is ready, when master B attempts to read the software message, then the master B read may bypass the master A write and, thus, master B may read stale data and, therefore, receive an incorrect message.

Some master peripherals (e.g., EDMA3 transfer controllers with TCCMOD=0) will always wait for the write to complete before signaling an interrupt to the system, thus avoiding this race condition. For masters that do not have a hardware specification of write-read ordering, it may be necessary to specify data ordering via software.

If master A does not wait for indication that a write is complete, it must perform the following workaround:

1. Perform the required write to DDR3 memory space.
2. Perform a dummy write to the DDR3 memory controller module ID and revision register.
3. Perform a dummy read to the DDR3 memory controller module ID and revision register.
4. Indicate to master B that the data is ready to be read after completion of the read in step 3. The completion of the read in step 3 ensures that the previous write was done.

7.11.2 DDR3 Memory Controller Electrical Data/Timing

The *DDR3 Implementation Guidelines* Application Report in 2.9 "Related Documentation from Texas Instruments" on page 66 specifies a complete DDR3 interface solution as well as a list of compatible DDR3 devices. The DDR3 electrical requirements are fully specified in the DDR3 Jedec Specification JESD79-3C. TI has performed the simulation and system characterization to ensure all DDR3 interface timings in this solution are met; therefore, no electrical data/timing information is supplied here for this interface.

Note—TI supports ***only*** designs that follow the board design guidelines outlined in the application report.

7.12 I²C Peripheral

The Inter-Integrated Circuit (I^2C) module provides an interface between DSP and other devices compliant with Philips Semiconductors Inter-IC bus (I^2C bus) specification version 2.1 and connected by way of an I^2C bus. External components attached to this 2-wire serial bus can transmit/receive up to 8-bit data to/from the SoC through the I^2C module.

7.12.1 I²C Device-Specific Information

The TMS320C6670 device includes an I^2C peripheral module. NOTE: when using the I^2C module, ensure there are external pullup resistors on the SDA and SCL pins.

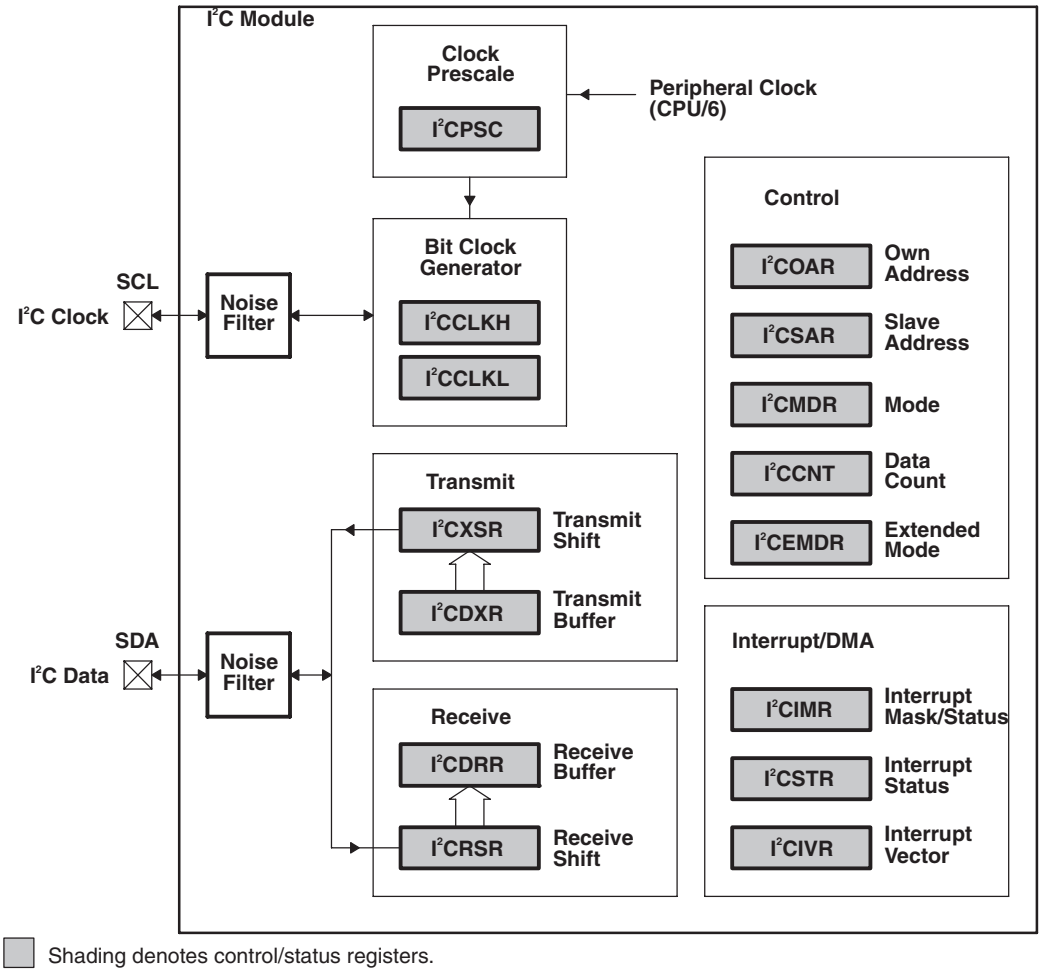
The I^2C modules on the C6670 may be used by the DSP to control local peripheral ICs (DACs, ADCs, etc.) or may be used to communicate with other controllers in a system or to implement a user interface.

The I^2C port supports:

- Compatibility with Philips I^2C specification revision 2.1 (January 2000)
- Fast mode up to 400 kbps (no fail-safe I/O buffers)
- Noise filter to remove noise of 50 ns or less
- 7-bit and 10-bit device addressing modes
- Multi-master (transmit/receive) and slave (transmit/receive) functionality
- Events: DMA, interrupt, or polling
- Slew-rate limited open-drain output buffers

Figure 7-35 shows a block diagram of the I2C module.

Figure 7-35 I2C Module Block Diagram



7.12.2 I2C Peripheral Register Description(s)

Table 7-65 I2C Registers (Part 1 of 2)

Hex Address Range	Acronym	Register Name
0253 0000	ICOAR	I2C own Address Register
0253 0004	ICIMR	I2C Interrupt Mask/status Register
0253 0008	ICSTR	I2C Interrupt Status Register
0253 000C	ICCLKL	I2C Clock Low-time Divider Register
0253 0010	ICCLKH	I2C Clock High-time Divider Register
0253 0014	ICCNT	I2C Data Count Register
0253 0018	ICDRR	I2C Data Receive Register
0253 001C	ICSAR	I2C Slave Address Register
0253 0020	ICDXR	I2C Data Transmit Register
0253 0024	ICMDR	I2C Mode Register
0253 0028	ICIVR	I2C Interrupt Vector Register
0253 002C	ICEMDR	I2C Extended Mode Register
0253 0030	ICPSC	I2C Prescaler Register

Table 7-65 I^2C Registers (Part 2 of 2)

Hex Address Range	Acronym	Register Name
0253 0034	ICPID1	I^2C Peripheral Identification Register 1 [value: 0x0000 0105]
0253 0038	ICPID2	I^2C Peripheral Identification Register 2 [value: 0x0000 0005]
0253 003C -0253 007F	-	Reserved
End of Table 7-65		

7.12.3 I²C Electrical Data/Timing

7.12.3.1 Inter-Integrated Circuits (I²C) Timing

Table 7-66 I²C Timing Requirements (1)
(see Figure 7-36)

No.			Standard Mode		Fast Mode		Units
			Min	Max	Min	Max	
1	$t_{c(SCL)}$	Cycle time, SCL	10		2.5		µs
2	$t_{su(SCLH-SDAL)}$	Setup time, SCL high before SDA low (for a repeated START condition)	4.7		0.6		µs
3	$t_{h(SDAL-SCLL)}$	Hold time, SCL low after SDA low (for a START and a repeated START condition)	4		0.6		µs
4	$t_{w(SCLL)}$	Pulse duration, SCL low	4.7		1.3		µs
5	$t_{w(SCLH)}$	Pulse duration, SCL high	4		0.6		µs
6	$t_{su(SDAV-SCLH)}$	Setup time, SDA valid before SCL high	250		100 (2)		ns
7	$t_{h(SCLL-SDAV)}$	Hold time, SDA valid after SCL low (for I²C bus devices)	0 (3)	3.45	0 (3)	0.9 (4)	µs
8	$t_{w(SDAH)}$	Pulse duration, SDA high between STOP and START conditions	4.7		1.3		µs
9	$t_{r(SDA)}$	Rise time, SDA		1000	$20 + 0.1C_b$ (5)	300	ns
10	$t_{r(SCL)}$	Rise time, SCL		1000	$20 + 0.1C_b$ (5)	300	ns
11	$t_{f(SDA)}$	Fall time, SDA		300	$20 + 0.1C_b$ (5)	300	ns
12	$t_{f(SCL)}$	Fall time, SCL		300	$20 + 0.1C_b$ (5)	300	ns
13	$t_{su(SCLH-SDAH)}$	Setup time, SCL high before SDA high (for STOP condition)	4		0.6		µs
14	$t_{w(SP)}$	Pulse duration, spike (must be suppressed)			0	50	ns
	C_b (5)	Capacitive load for each bus line		400		400	pF
End of Table 7-66							

1 The I²C pins SDA and SCL do not feature fail-safe I/O buffers. These pins could potentially draw current when the device is powered down

2 A Fast-mode I²C-bus™ device can be used in a Standard-mode I²C-bus™ system, but the requirement tsu(SDA-SCLH) ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line t_r max + $t_{su(SDA-SCLH)}$ = 1000 + 250 = 1250 ns (according to the Standard-mode I²C-Bus Specification) before the SCL line is released.

3 A device must internally provide a hold time of at least 300 ns for the SDA signal (referred to the V_{IHmin} of the SCL signal) to bridge the undefined region of the falling edge of SCL.

4 The maximum $t_{h(SDA-SCLL)}$ has to be met only if the device does not stretch the low period [$t_{w(SCLL)}$] of the SCL signal.

5 C_b = total capacitance of one bus line in pF. If mixed with HS-mode devices, faster fall-times are allowed.

Figure 7-36 I²C Receive Timings

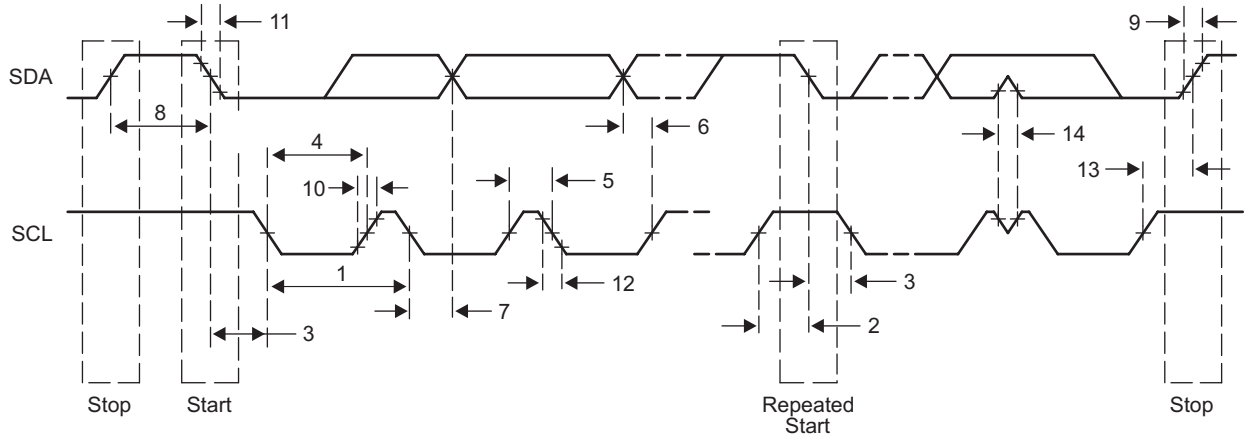


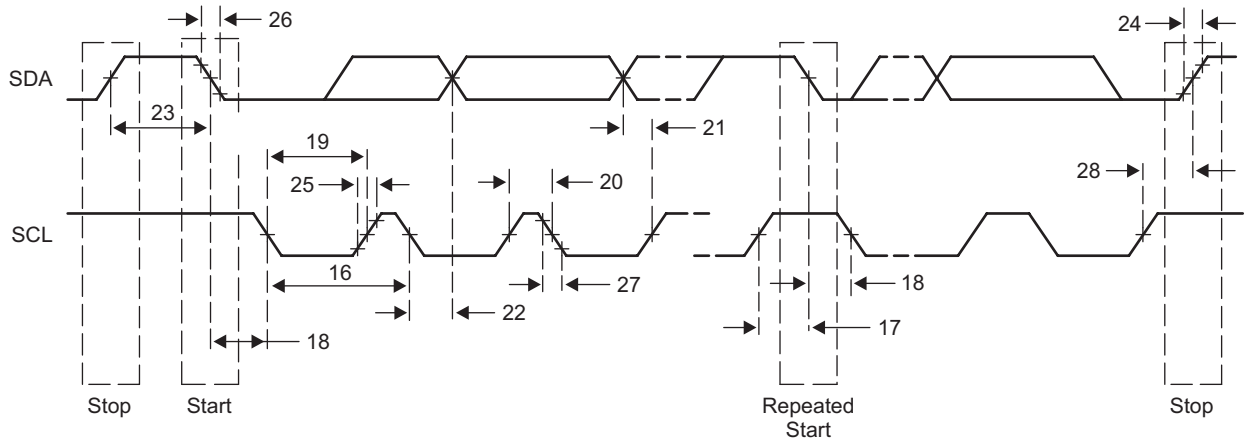
Table 7-67 I^2C Switching Characteristics [1]
(see Figure 7-37)

No.	Parameter		Standard Mode Min	Standard Mode Max	Fast Mode Min	Fast Mode Max	Unit
16	$t_{c(SCL)}$	Cycle time, SCL	10		2.5		ms
17	$t_{su(SCLH-SDAL)}$	Setup time, SCL high to SDA low (for a repeated START condition)	4.7		0.6		ms
18	$t_{h(SDAL-SCLL)}$	Hold time, SDA low after SCL low (for a START and a repeated START condition)	4		0.6		ms
19	$t_{w(SCLL)}$	Pulse duration, SCL low	4.7		1.3		ms
20	$t_{w(SCLH)}$	Pulse duration, SCL high	4		0.6		ms
21	$t_{d(SDAV-SDLH)}$	Delay time, SDA valid to SCL high	250		100		ns
22	$t_{v(SDLL-SDAV)}$	Valid time, SDA valid after SCL low (for I^2C bus devices)	0		0	0.9	ms
23	$t_{w(SDAH)}$	Pulse duration, SDA high between STOP and START conditions	4.7		1.3		ms
24	$t_{r(SDA)}$	Rise time, SDA		1000	$20 + 0.1C_b$ [1]	300	ns
25	$t_{r(SCL)}$	Rise time, SCL		1000	$20 + 0.1C_b$ [1]	300	ns
26	$t_{f(SDA)}$	Fall time, SDA		300	$20 + 0.1C_b$ [1]	300	ns
27	$t_{f(SCL)}$	Fall time, SCL		300	$20 + 0.1C_b$ [1]	300	ns
28	$t_{d(SCLH-SDAH)}$	Delay time, SCL high to SDA high (for STOP condition)	4		0.6		ms
	C_p	Capacitance for each I^2C pin		10		10	pF

End of Table 7-67

1 C_b = total capacitance of one bus line in pF. If mixed with HS-mode devices, faster fall-times are allowed.

Figure 7-37 I^2C Transmit Timings



7.13 SPI Peripheral

The Serial Peripheral Interconnect (SPI) module provides an interface between the DSP and other SPI-compliant devices. The primary intent of this interface is to allow for connection to a SPI ROM for boot. The SPI module on C6670 is supported only in Master mode. Additional chip-level components can also be included, such as temperature sensors or an I/O expander.

7.13.1 SPI Electrical Data/Timing

Table 7-68 SPI Timing Requirements
See Figure 7-38)

No.			Min	Max	Unit
Master Mode Timing Diagrams — Base Timings for 3 Pin Mode					
7	tsu(SOMI-SPC)	Input setup time, SPIx_SOMI valid before receive edge of SPIx_CLk. Polarity = 0 Phase = 0	2		ns
7	tsu(SOMI-SPC)	Input setup time, SPIx_SOMI valid before receive edge of SPIx_CLk. Polarity = 0 Phase = 1	2		ns
7	tsu(SOMI-SPC)	Input setup time, SPIx_SOMI valid before receive edge of SPIx_CLk. Polarity = 1 Phase = 0	2		ns
7	tsu(SOMI-SPC)	Input setup time, SPIx_SOMI valid before receive edge of SPIx_CLk. Polarity = 1 Phase = 1	2		ns
8	th(SPC-SOMI)	Input hold time, SPIx_SOMI valid after receive edge of SPIx_CLK. Polarity = 0 Phase = 0	5		ns
8	th(SPC-SOMI)	Input hold time, SPIx_SOMI valid after receive edge of SPIx_CLK. Polarity = 0 Phase = 1	5		ns
8	th(SPC-SOMI)	Input hold time, SPIx_SOMI valid after receive edge of SPIx_CLK. Polarity = 1 Phase = 0	5		ns
8	th(SPC-SOMI)	Input hold time, SPIx_SOMI valid after receive edge of SPIx_CLK. Polarity = 1 Phase = 1	5		ns
End of Table 7-68					

Table 7-69 SPI Switching Characteristics (Part 1 of 2)
(See Figure 7-38 and Figure 7-39)

No.	Parameter		Min	Max	Unit
Master Mode Timing Diagrams — Base Timings for 3 Pin Mode					
1	tc(SPC)	Cycle time, SPIx_CLK, all master modes	3*P2 (1)		ns
2	tw(SPCH)	Pulse width high, SPIx_CLK, all master modes	0.5*(3*P2) - 1		ns
3	tw(SPCL)	Pulse width low, SPIx_CLK, all master modes	0.5*(3*P2) - 1		ns
4	td(SIMO-SPC)	Setup (Delay), initial data bit valid on SPIx_SIMO to initial edge on SPIx_CLK. Polarity = 0, Phase = 0.		5	ns
4	td(SIMO-SPC)	Setup (Delay), initial data bit valid on SPIx_SIMO to initial edge on SPIx_CLK. Polarity = 0, Phase = 1.		5	ns
4	td(SIMO-SPC)	Setup (Delay), initial data bit valid on SPIx_SIMO to initial edge on SPIx_CLK Polarity = 1, Phase = 0		5	ns
4	td(SIMO-SPC)	Setup (Delay), initial data bit valid on SPIx_SIMO to initial edge on SPIx_CLK Polarity = 1, Phase = 1		5	ns
5	td(SPC-SIMO)	Setup (Delay), subsequent data bits valid on SPIx_SIMO to initial edge on SPIx_CLK. Polarity = 0 Phase = 0		2	ns
5	td(SPC-SIMO)	Setup (Delay), subsequent data bits valid on SPIx_SIMO to initial edge on SPIx_CLK Polarity = 0 Phase = 1		2	ns
5	td(SPC-SIMO)	Setup (Delay), subsequent data bits valid on SPIx_SIMO to initial edge on SPIx_CLK Polarity = 1 Phase = 0		2	ns
5	td(SPC-SIMO)	Setup (Delay), subsequent data bits valid on SPIx_SIMO to initial edge on SPIx_CLK Polarity = 1 Phase = 1		2	ns
6	toh(SPC-SIMO)	Output hold time, SPIx_SIMO valid after receive edge of SPIx_CLK except for final bit. Polarity = 0 Phase = 0	0.5*tc - 2		ns
6	toh(SPC-SIMO)	Output hold time, SPIx_SIMO valid after receive edge of SPIx_CLK except for final bit. Polarity = 0 Phase = 1	0.5*tc - 2		ns
6	toh(SPC-SIMO)	Output hold time, SPIx_SIMO valid after receive edge of SPIx_CLK except for final bit. Polarity = 1 Phase = 0	0.5*tc - 2		ns

Table 7-69 SPI Switching Characteristics (Part 2 of 2)
(See Figure 7-38 and Figure 7-39)

No.	Parameter		Min	Max	Unit
6	toh(SPC-SIMO)	Output hold time, SPIx_SIMO valid after receive edge of SPIx_CLK except for final bit. Polarity = 1 Phase = 1	0.5*tc - 2		ns
Additional SPI Master Timings — 4 Pin Mode with Chip Select Option					
19	td(SCS-SPC)	Delay from SPIx_SCS\ active to first SPIx_CLK. Polarity = 0 Phase = 0	2*P2 - 5	2*P2 + 5	ns
19	td(SCS-SPC)	Delay from SPIx_SCS\ active to first SPIx_CLK. Polarity = 0 Phase = 1	0.5*tc + (2*P2) - 5	0.5*tc + (2*P2) + 5	ns
19	td(SCS-SPC)	Delay from SPIx_SCS\ active to first SPIx_CLK. Polarity = 1 Phase = 0	2*P2 - 5	2*P2 + 5	ns
19	td(SCS-SPC)	Delay from SPIx_SCS\ active to first SPIx_CLK. Polarity = 1 Phase = 1	0.5*tc + (2*P2) - 5	0.5*tc + (2*P2) + 5	ns
20	td(SPC-SCS)	Delay from final SPIx_CLK edge to master deasserting SPIx_SCS\. Polarity = 0 Phase = 0	1*P2 - 5	1*P2 + 5	ns
20	td(SPC-SCS)	Delay from final SPIx_CLK edge to master deasserting SPIx_SCS\. Polarity = 0 Phase = 1	0.5*tc + (1*P2) - 5	0.5*tc + (1*P2) + 5	ns
20	td(SPC-SCS)	Delay from final SPIx_CLK edge to master deasserting SPIx_SCS\. Polarity = 1 Phase = 0	1*P2 - 5	1*P2 + 5	ns
20	td(SPC-SCS)	Delay from final SPIx_CLK edge to master deasserting SPIx_SCS\. Polarity = 1 Phase = 1	0.5*tc + (1*P2) - 5	0.5*tc + (1*P2) + 5	ns
	tw(SCSH)	Minimum inactive time on SPIx_SCS\ pin between two transfers when SPIx_SCS\ is not held using the CSHOLD feature.	2*P2 - 5		ns
End of Table 7-69					

1 P2=1/SYSCLK7

Figure 7-38 SPI Master Mode Timing Diagrams — Base Timings for 3-Pin Mode

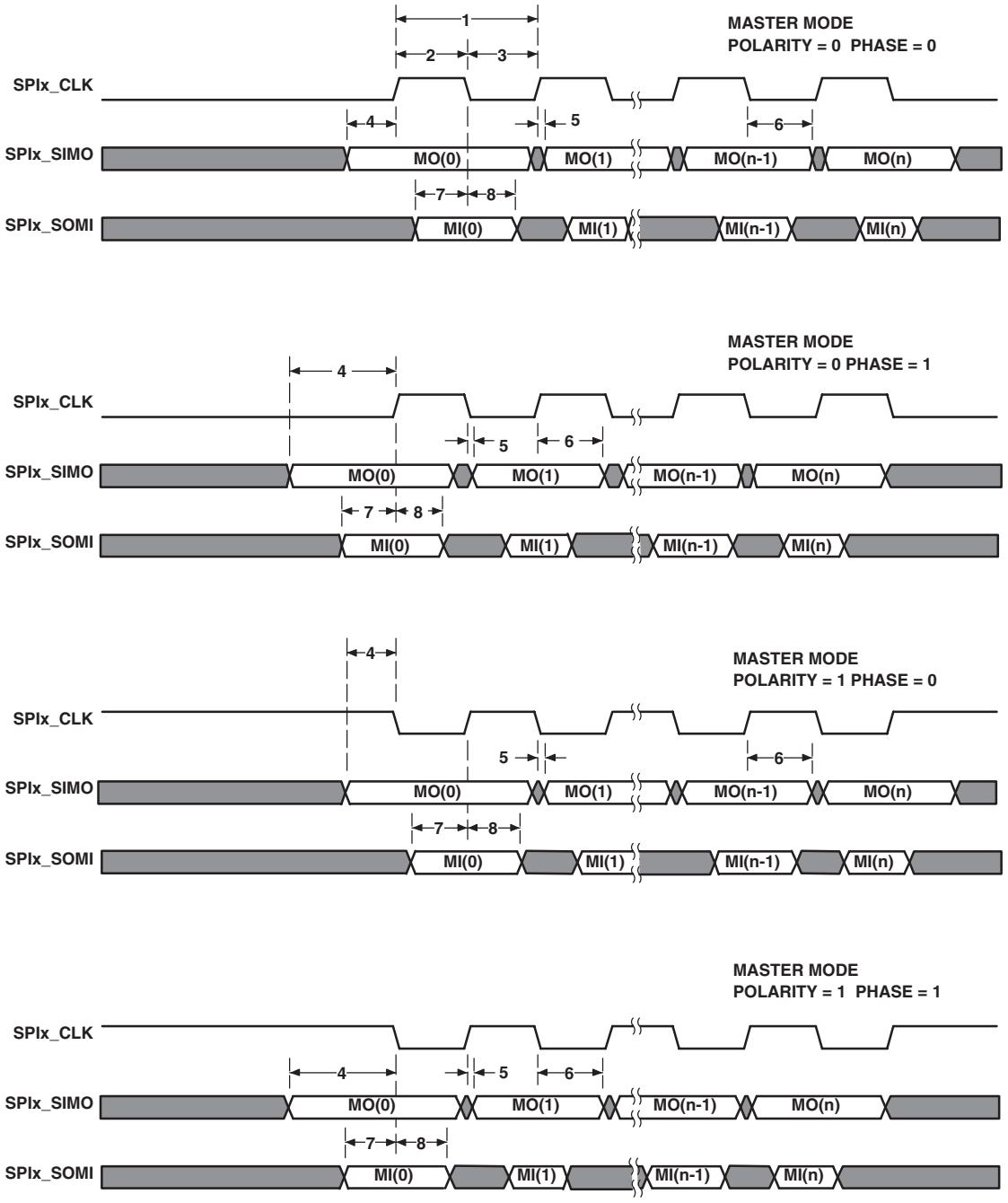
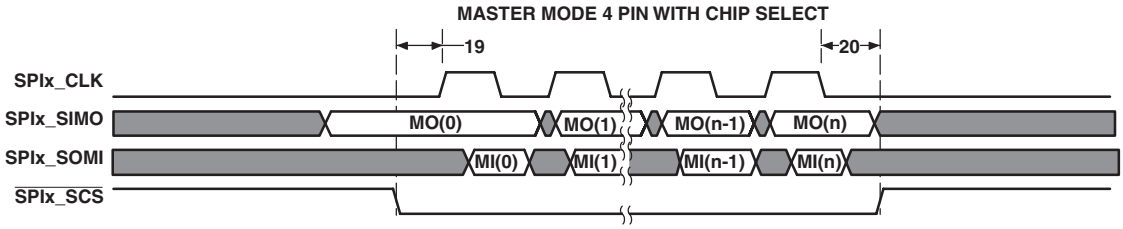


Figure 7-39 SPI Additional Timings for 4-Pin Master Mode with Chip Select Option



7.14 HyperLink Peripheral

The TMS320C6670 includes the HyperLink for companion chip/die interfaces. This is a four-lane SerDes interface designed to operate up to 12.5 Gbps per lane from pin-to-pin. The interface is used to connect with external accelerators that are manufactured using TI libraries. The Hyperbridge links must be connected with DC coupling.

The interface includes the serial station management interfaces used to send power management and flow messages between devices. This consists of four LVCMOS inputs and four LVCMOS outputs configured as two 2-wire output buses and two 2-wire input buses. Each 2-wire bus includes a data signal and a clock signal.

Table 7-70 HyperLink Peripheral Timing Requirements
(see Figure 7-40, Figure 7-41 and Figure 7-42)

No.			Min	Max	Unit
FL Interface					
1	tc(MCMTXFLCLK)	Clock period - MCMTXFLCLK (C1)	5.75		ns
2	tw(MCMTXFLCLKH)	High pulse width - MCMTXFLCLK	0.4*C1	0.6*C1	ns
3	tw(MCMTXFLCLKL)	Low pulse width - MCMTXFLCLK	0.4*C1	0.6*C1	ns
6	tsu(MCMTXFLDAT-MCMTXFLCLKH)	Setup time - MCMTXFLDAT valid before MCMTXFLCLK high	1		ns
7	th(MCMTXFLCLKH-MCMTXFLDAT)	Hold time - MCMTXFLDAT valid after MCMTXFLCLK high	1		ns
6	tsu(MCMTXFLDAT-MCMTXFLCLKL)	Setup time - MCMTXFLDAT valid before MCMTXFLCLK low	1		ns
7	th(MCMTXFLCLKL-MCMTXFLDAT)	Hold time - MCMTXFLDAT valid after MCMTXFLCLK low	1		ns
PM Interface					
1	tc(MCMRXPMCLK)	Clock period - MCMRXPMCLK (C3)	5.75		ns
2	tw(MCMRXPMCLK)	High pulse width - MCMRXPMCLK	0.4*C3	0.6*C3	ns
3	tw(MCMRXPMCLK)	Low pulse width - MCMRXPMCLK	0.4*C3	0.6*C3	ns
6	tsu(MCMRXPMDAT-MCMRXPMCLKH)	Setup time - MCMRXPMDAT valid before MCMRXPMCLK high	1		ns
7	th(MCMRXPMCLKH-MCMRXPMDAT)	Hold time - MCMRXPMDAT valid after MCMRXPMCLK high	1		ns
6	tsu(MCMRXPMDAT-MCMRXPMCLKL)	Setup time - MCMRXPMDAT valid before MCMRXPMCLK low	1		ns
7	th(MCMRXPMCLKL-MCMRXPMDAT)	Hold time - MCMRXPMDAT valid after MCMRXPMCLK low	1		ns
End of Table 7-70					

Table 7-71 HyperLink Peripheral Switching Characteristics (Part 1 of 2)
(see Figure 7-40, Figure 7-41 and Figure 7-42)

No.	Parameter		Min	Max	Unit
FL Interface					
1	tc(MCMRXFLCLK)	Clock period - MCMRXFLCLK (C2)	6.4		ns
2	tw(MCMRXFLCLKH)	High pulse width - MCMRXFLCLK	0.4*C2	0.6*C2	ns
3	tw(MCMRXFLCLKL)	Low pulse width - MCMRXFLCLK	0.4*C2	0.6*C2	ns
4	tosu(MCMRXFLDAT-MCMRXFLCLKH)	Setup time - MCMRXFLDAT valid before MCMRXFLCLK high	0.25*C2-0.4		ns
5	toh(MCMRXFLCLKH-MCMRXFLDAT)	Hold time - MCMRXFLDAT valid after MCMRXFLCLK high	0.25*C2-0.4		ns
4	tosu(MCMRXFLDAT-MCMRXFLCLKL)	Setup time - MCMRXFLDAT valid before MCMRXFLCLK low	0.25*C2-0.4		ns
5	toh(MCMRXFLCLKL-MCMRXFLDAT)	Hold time - MCMRXFLDAT valid after MCMRXFLCLK low	0.25*C2-0.4		ns
PM Interface					
1	tc(MCMTXPMCLK)	Clock period - MCMTXPMCLK (C4)	6.4		ns
2	tw(MCMTXPMCLK)	High pulse width - MCMTXPMCLK	0.4*C4	0.6*C4	ns
3	tw(MCMTXPMCLK)	Low pulse width - MCMTXPMCLK	0.4*C4	0.6*C4	ns
4	tosu(MCMTXPMDAT-MCMTXPMCLKH)	Setup time - MCMTXPMDAT valid before MCMTXPMCLK high	0.25*C2-0.4		ns
5	toh(MCMTXPMCLKH-MCMTXPMDAT)	Hold time - MCMTXPMDAT valid after MCMTXPMCLK high	0.25*C2-0.4		ns

Table 7-71 HyperLink Peripheral Switching Characteristics (Part 2 of 2)
(see Figure 7-40, Figure 7-41 and Figure 7-42)

No.	Parameter		Min	Max	Unit
4	tosu(MCMTXPMDAT-MCMTXPMCLKL)	Setup time - MCMTXPMDAT valid before MCMTXPMCLK low	0.25*C2-0.4		ns
5	toh(MCMTXPMCLKL-MCMTXPMDAT)	Hold time - MCMTXPMDAT valid after MCMTXPMCLK low	0.25*C2-0.4		ns
End of Table 7-71					

Figure 7-40 HyperLink Station Management Clock Timing

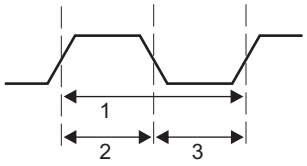
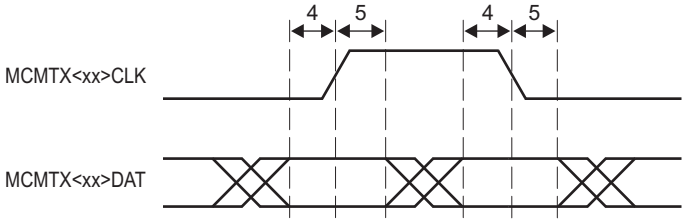
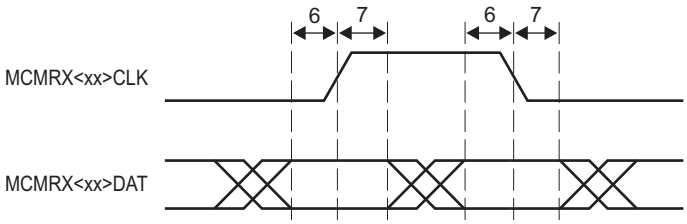


Figure 7-41 HyperLink Station Management Transmit Timing



<xx> represents the interface that is being used: PM or FL

Figure 7-42 HyperLink Station Management Receive Timing



<xx> represents the interface that is being used: PM or FL

7.15 UART Peripheral

The universal asynchronous receiver/transmitter (UART) module provides an interface between the DSP and a UART terminal interface or other UART-based peripheral. The UART is based on the industry standard TL16C550 asynchronous communications element, which, in turn, is a functional upgrade of the TL16C450. Functionally similar to the TL16C450 on power up (single character or TL16C450 mode), the UART can be placed in an alternate FIFO (TL16C550) mode. This relieves the DSP of excessive software overhead by buffering received and transmitted characters. The receiver and transmitter FIFOs store up to 16 bytes including three additional bits of error status per byte for the receiver FIFO.

The UART performs serial-to-parallel conversions on data received from a peripheral device and parallel-to-serial conversion on data received from the DSP. The DSP can read the UART status at any time. The UART includes control capability and a processor interrupt system that can be tailored to minimize software management of the communications link. For more information on UART, see the *Universal Asynchronous Receiver/Transmitter (UART) for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

Table 7-72 UART Timing Requirements
(see Figure 7-43 and Figure 7-44)

No.			Min	Max	Unit
Receive Timing					
4	tw(RXSTART)	Pulse width, receive start bit	0.96U [1]	1.05U	ns
5	tw(RXH)	Pulse width, receive data/parity bit high	0.96U	1.05U	ns
5	tw(RXL)	Pulse width, receive data/parity bit low	0.96U	1.05U	ns
6	tw(RXSTOP1)	Pulse width, receive stop bit 1	0.96U	1.05U	ns
6	tw(RXSTOP15)	Pulse width, receive stop bit 1.5	0.96U	1.05U	ns
6	tw(RXSTOP2)	Pulse width, receive stop bit 2	0.96U	1.05U	ns
Autoflow Timing Requirements					
8	td(CTSL-TX)	Delay time, CTS asserted to START bit transmit	P [2]	5P	ns
End of Table 7-72					

1 U = UART baud time = 1/programmed baud rate
2 P = 1/SYSCLK7

Figure 7-43 UART Receive Timing Waveform

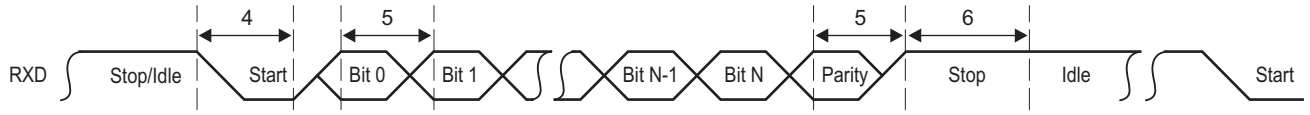


Figure 7-44 UART CTS (Clear-to-Send Input) — Autoflow Timing Waveform

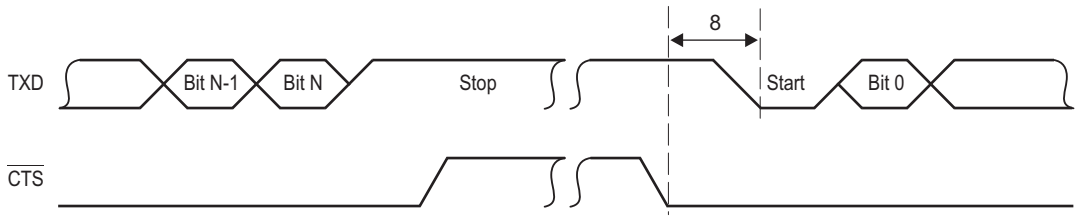


Table 7-73 UART Switching Characteristics
(See Figure 7-45 and Figure 7-46)

No.	Parameter		Min	Max	Unit
Transmit Timing					
1	tw(TXSTART)	Pulse width, transmit start bit	U (1) - 2	U + 2	ns
2	tw(TXH)	Pulse width, transmit data/parity bit high	U - 2	U + 2	ns
2	tw(TXL)	Pulse width, transmit data/parity bit low	U - 2	U + 2	ns
3	tw(TXSTOP1)	Pulse width, transmit stop bit 1	U - 2	U + 2	ns
3	tw(TXSTOP15)	Pulse width, transmit stop bit 1.5	1.5 * (U - 2)	1.5 * ('U + 2)	ns
3	tw(TXSTOP2)	Pulse width, transmit stop bit 2	2 * (U - 2)	2 * ('U + 2)	ns
Autoflow Timing Requirements					
7	td(RX-RTSH)	Delay time, STOP bit received to RTS deasserted	P (2)	5P	ns
End of Table 7-73					

1 U = UART baud time = 1/programmed baud rate
2 P = 1/SYSCLK7

Figure 7-45 UART Transmit Timing Waveform

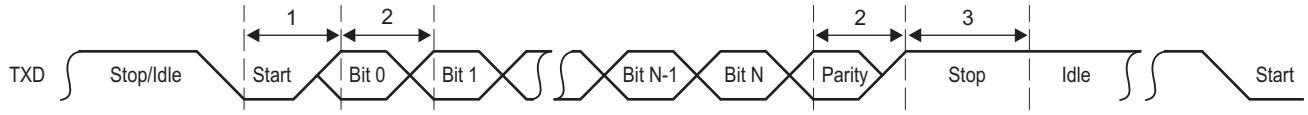
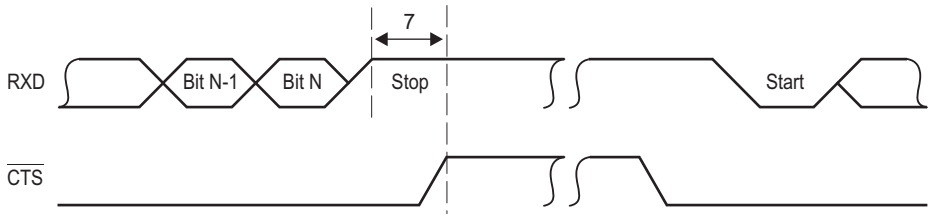


Figure 7-46 UART RTS (Request-to-Send Output) – Autoflow Timing Waveform



7.16 PCIe Peripheral

The 2-lane PCI express (PCIe) module on TMS320C6670 provides an interface between the SoC and other PCIe-compliant devices. The PCI express module provides low pin-count, high-reliability, and high-speed data transfer at rates of 5.0 Gbps per lane on the serial links. For more information, see the *Peripheral Component Interconnect Express (PCIe) for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

7.17 Packet Accelerator

The Packet Accelerator provides L2 to L4 classification functionalities. It supports classification for Ethernet, VLAN, MPLS over Ethernet, IPv4/6, GRE over IP, and other session identification over IP such as TCP and UDP ports. It maintains 8k multiple-in, multiple-out hardware queues. It also provides checksum capability as well as some QoS capabilities. It enables a single IP address to be used for a multi-core device. It can process up to 1.5 Mpps. The Packet Accelerator is coupled with the Network Coprocessor. For more information, see the *Packet Accelerator (PA) for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

7.18 Security Accelerator

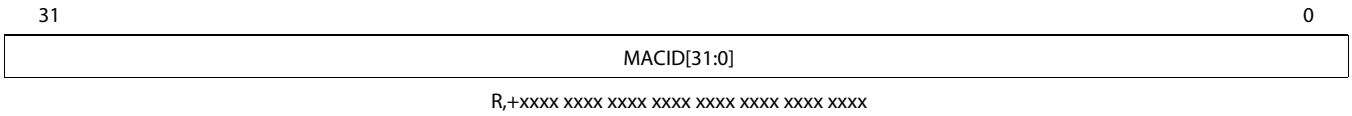
The Security Accelerator provides wire-speed processing on 1-Gbps Ethernet traffic on IPSec, SRTP, and 3GPP Air interface security protocols. It functions on the packet level with the packet and the associated security context being one of these above three types. The Security Accelerator is coupled with network coprocessor, and receives the packet descriptor containing the security context in the buffer descriptor, and the data to be encrypted/decrypted in the linked buffer descriptor. For more information, see the Security Accelerator (SA) *for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

7.19 Gigabit Ethernet (GbE) Switch Subsystem

The gigabit Ethernet (GbE) switch subsystem provide an efficient interface between the TMS320C6670 SoC and the networked community. The EMAC supports 10Base-T (10 Mbits/second [Mbps]), and 100BaseTX (100 Mbps), in half- or full-duplex mode, and 1000BaseT (1000 Mbps) in full-duplex mode, with hardware flow control and quality-of-service (QOS) support. The GbE switch subsystem is coupled with network coprocessor. For more information, see the *Gigabit Ethernet (GbE) Switch Subsystem for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

Each device has a unique MAC address. There are two registers to hold these values, MACID1 (0x02620110) and MACID2 (0x02620114). All bits of these registers are defined as follows:

Figure 7-47 MACID1 Register

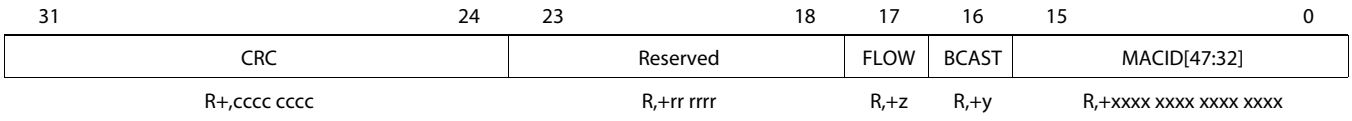


Legend: R = Read only; -x, value is indeterminate

Table 7-74 MACID1 Register Field Descriptions

Bit	Field	Description
31-0	MAC ID[31-0]	MAC ID. A range will be assigned to this device. Each device will consume only one MAC address.
End of Table 7-74		

Figure 7-48 MACID2 Register



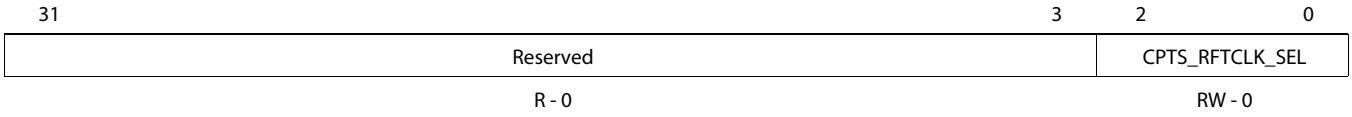
Legend: R = Read only; -x, value is indeterminate

Table 7-75 MACID2 Register Field Descriptions

Bit	Field	Description
31-24	Reserved	Variable
23-18	Reserved	000000
17	FLOW	MAC Flow Control 0 = Off 1 = On
16	BCAST	Default m/b-cast reception 0 = Broadcast 1 = Disabled
15-0	MAC ID[47-0]	MAC ID. A range will be assigned to this device. Each device will consume only one MAC address.
End of Table 7-75		

There is a Central Processor Time Synchronization (CPTS) submodule in the Ethernet switch module that can be used for time synchronization. Programming this register selects the clock source for the CPTS_RCLK. Please see the see the *Gigabit Ethernet (GbE) Switch Subsystem for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66 for the register address and other details about the time synchronization module. The register CPTS_RFTCLK_SEL for reference clock selection of time synchronization submodule is shown in Figure 7-49.

Figure 7-49 RFTCLK Select Register (CPTS_RFTCLK_SEL)



Legend: R = Read only; -x, value is indeterminate

Table 7-76 RFTCLK Select Register Field Descriptions

Bit	Field	Description
31-3	Reserved	Reserved. Read as zero.
2-0	CPTS_RFTCLK_SEL	Reference clock select. This signal is used to control an external multiplexer that selects one of 8 clocks for time sync reference (RFTCLK). This CPTS_RFTCLK_SEL value can be written only when the CPTS_EN bit is cleared to 0 in the TS_CTL register. 000 = SYSCLK2 001 = SYSCLK3 010 = TIMI0 011 = TIMI1 1xx = Reserved
End of Table 7-76		

7.20 Management Data Input/Output (MDIO)

The management data input/output (MDIO) module implements the 802.3 serial management interface to interrogate and controls up to 32 Ethernet PHY(s) connected to the device, using a shared two-wire bus. Application software uses the MDIO module to configure the auto-negotiation parameters of each PHY attached to the EMAC, retrieve the negotiation results, and configure required parameters in the to gigabit Ethernet (GbE) switch subsystem for correct operation. The module is designed to allow almost transparent operation of the MDIO interface, with very little attention from the CorePac. For more information, see the *Gigabit Ethernet (GbE) Switch Subsystem for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

Table 7-77 MDIO Timing Requirements
(see Figure 7-50)

No.			Min	Max	Unit
1	tc(MDCLK)	Cycle time, MDCLK	400		ns
2	tw(MDCLKH)	Pulse duration, MDCLK high	180		ns
3	tw(MDCLKL)	Pulse duration, MDCLK low	180		ns
4	tsu(MDIO-MDCLKH)	Setup time, MDIO data input valid before MDCLK high	10		ns
5	th(MDCLKH-MDIO)	Hold time, MDIO data input valid after MDCLK high	10		ns
	tt(MDCLK)	Transition time, MDCLK		5	ns
End of Table 7-77					

Figure 7-50 MDIO Input Timing

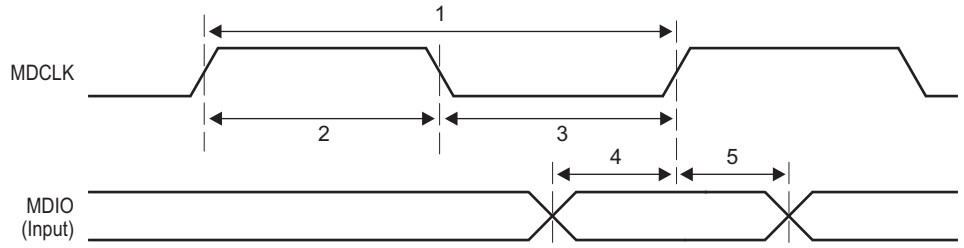
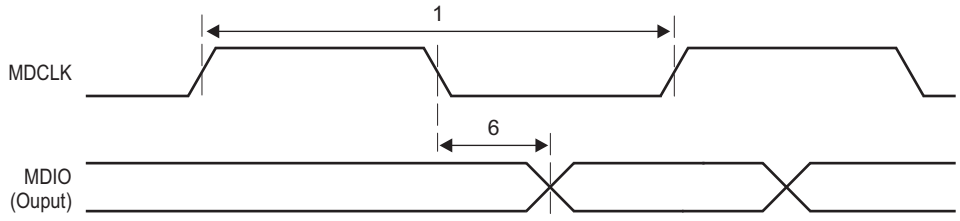


Table 7-78 MDIO Switching Characteristics
(see Figure 7-51)

No.	Parameter		Min	Max	Unit
6	td(MDCLKL-MDIO)	Delay time, MDCLK low to MDIO data output valid		100	ns
End of Table 7-78					

Figure 7-51 MDIO Output Timing



7.21 Timers

The timers can be used to time events, count events, generate pulses, interrupt the CPU, and send synchronization events to the EDMA3 channel controller.

7.21.1 Timers Device-Specific Information

The TMS320C6670 device has eight 64-bit timers in total. Of which Timer0 through Timer3 are dedicated to each of the four CorePacs as a watchdog timer and can also be used as general-purpose timers. Each of other four timers can also be configured as a general-purpose timer only, with each timer programmed as a 64-bit timer or as two separate 32-bit timers.

When operating in 64-bit mode, the timer counts either VBUS clock cycles or input (TINPLx) pulses (rising edge) and generates an output pulse/waveform (TOUTLx) plus an internal event (TINTLx) on a software-programmable period. When operating in 32-bit mode, the timer is split into two independent 32-bit timers. Each timer is made up of two 32-bit counters: a high counter and a low counter. The timer pins, TINPLx and TOUTLx are connected to the low counter. The timer pins, TINPHx and TOUTHx are connected to the high counter.

When operating in watchdog mode, the timer counts down to 0 and generates an event. It is a requirement that software writes to the timer before the count expires, after which the count begins again. If the count ever reaches 0, the timer event output is asserted. Reset initiated by a watchdog timer can be set by programming "Reset Type Status Register (RSTYPE)" on page 135 and the type of reset initiated can set by programming "Reset Configuration Register (RSTCFG)" on page 136. For more information, see the *64-bit Timer (Timer 64) for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

7.21.2 Timers Electrical Data/Timing

The tables and figures below describe the timing requirements and switching characteristics of Timer0 - Timer7.

Table 7-79 Timer Input Timing Requirements [1]
(see Figure 7-52)

No.			Min	Max	Unit
1	$t_{w(TINPH)}$	Pulse duration, high	12C		ns
2	$t_{w(TINPL)}$	Pulse duration, low	12C		ns
End of Table 7-79					

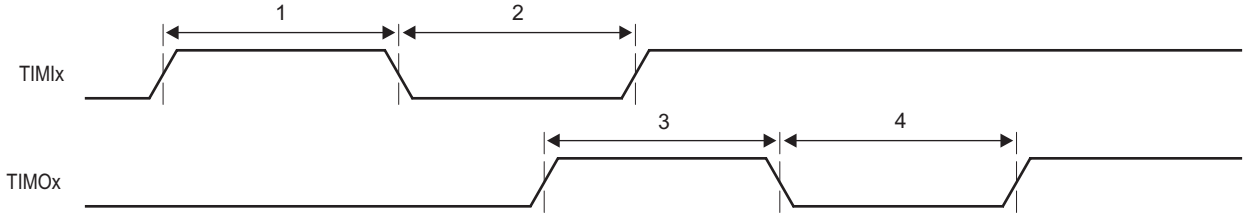
1 C = 1/SYSCLK1 clock frequency in ns

Table 7-80 Timer Output Switching Characteristics [1]
(see Figure 7-52)

No.	Parameter		Min	Max	Unit
3	$t_{w(TOUTH)}$	Pulse duration, high	12C - 3		ns
4	$t_{w(TOUTL)}$	Pulse duration, low	12C - 3		ns
End of Table 7-80					

1 C = 1/SYSCLK1 clock frequency in ns.

Figure 7-52 Timer Timing



7.22 Rake Search Accelerator (RSA)

There are four Rake Search Accelerators (RSAs) on the device. CorePac1 and CorePac2 each have one set of directly-connected RSA pairs. The RSA is an extension of the C66x CPU. The CPU performs send/receive to the RSAs via the .L and .S functional units.

7.23 Enhanced Viterbi-Decoder Coprocessor (VCP2)

The device has four high-performance embedded Viterbi Decoder Coprocessors (VCP2) that significantly speed up channel-decoding operations on-chip. Each VCP2, operating at CPU clock divided-by-3, can decode more than 694 7.95-Kbps adaptive multi-rate (AMR) [K = 9, R = 1/3] voice channels. The VCP2 supports constraint lengths K = 5, 6, 7, 8, and 9, rates R = 3/4, 1/2, 1/3, 1/4, and 1/5, and flexible polynomials, while generating hard decisions or soft decisions. Communications between the VCP2 and the CPU are carried out through the EDMA3 controller. The VCP2 supports:

- Unlimited frame sizes
- Code rates 3/4, 1/2, 1/3, 1/4, and 1/5
- Constraint lengths 5, 6, 7, 8, and 9
- Programmable encoder polynomials
- Programmable reliability and convergence lengths
- Hard and soft decoded decisions
- Tail and convergent modes
- Yamamoto logic
- Tail biting logic
- Various input and output FIFO lengths

For more information, see the *Viterbi Coprocessor (VCP2) for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

7.24 Turbo Decoder Coprocessor (TCP3d)

The C6670 has two high-performance embedded Turbo-Decoder Coprocessors (TCP3d) that significantly speed up channel-decoding operations on-chip for WCDMA, HSPA, HSPA+, TD-SCDMA, LTE, and WiMAX. Operating at CPU clock divided-by-2, the TCP3d is capable of processing data channels at a throughput of >100 Mbps. For more information, see the *Turbo Decoder Coprocessor 3 (TCP3d) for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

7.25 Turbo Encoder Coprocessor (TCP3e)

The C6670 has a high-performance embedded Turbo-Encoder Coprocessor (TCP3e) that significantly speeds up channel-encoding operations on-chip for WCDMA, HSPA, HSPA+, TD-SCDMA, LTE, and WiMAX. Operating at CPU clock divided-by-3, the TCP3e is capable of processing data channels at a throughput of >200 Mbps. For more information, see the *Turbo Encoder Coprocessor 3 (TCP3e) for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

7.26 Bit Rate Coprocessor (BCP)

The BCP is a hardware accelerator for wireless infrastructure. It performs most of the uplink and downlink layer 1 bit processing for 3G and 4G wireless standards. It supports LTE, FDD WCDMA, TD-SCDMA, and WiMAX 802.16-2009 standards. It supports various downlink processing blocks like CRC attachment, turbo encoding, rate matching, code block concatenation, scrambling, and modulation. It supports various uplink processing blocks like soft slicer, de-scrambler, de-concatenation, rate de-matching and LLR combining. For more information, see the Bit Coprocessor (BCP) for KeyStone Devices User Guide in 2.9 "Related Documentation from Texas Instruments" on page 66.

7.27 Serial RapidIO (SRIO) Port

The SRIO port on the device is a high-performance, low pin-count SerDes interconnect. The use of the RapidIO interconnect in a baseband board design can create a homogeneous interconnect environment, providing connectivity and control among the components. RapidIO is based on the memory and device addressing concepts of processor buses in which the transaction processing is managed completely by hardware. This enables the RapidIO interconnect to lower the system cost by providing lower latency, reduced overhead of packet data processing, and higher system bandwidth, all of which are key for wireless interfaces. For more information, see the *Serial RapidIO (SRIO) for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

7.28 General-Purpose Input/Output (GPIO)

7.28.1 GPIO Device-Specific Information

On the TMS320C6670, the GPIO peripheral pins GP[15:0] are also used to latch configuration pins. For more detailed information on device/peripheral configuration and the C6670 device pin muxing, see "Device Configuration" on page 67.

7.28.2 GPIO Electrical Data/Timing

Table 7-81 GPIO Input Timing Requirements (1)
(see Figure 7-53)

No.			Min	Max	Unit
1	$t_{w(GPOH)}$	Pulse duration, GPOx high	12C		ns
2	$t_{w(GPOL)}$	Pulse duration, GPOx low	12C		ns
End of Table 7-81					

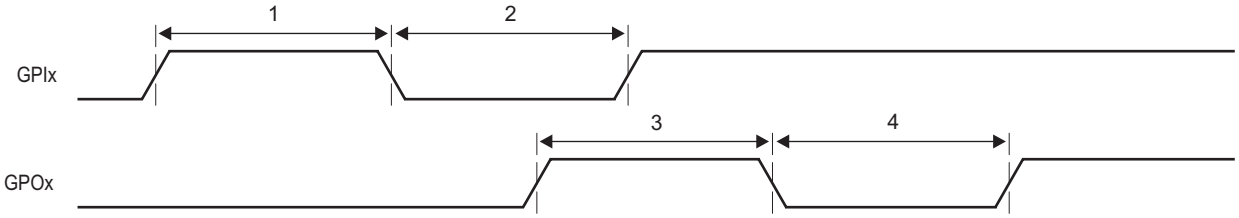
1 C = 1/SYSCLK1 clock frequency in ns

Table 7-82 GPIO Output Switching Characteristics (1)
(see Figure 7-53)

No.	Parameter		Min	Max	Unit
3	$t_{w(GPOH)}$	Pulse duration, GPOx high	36C - 8		ns
4	$t_{w(GPOL)}$	Pulse duration, GPOx low	36C - 8		ns
End of Table 7-82					

1 C = 1/SYSCLK1 clock frequency in ns

Figure 7-53 GPIO Timing



7.29 Semaphore2

The device contains an enhanced Semaphore module for the management of shared resources of the CorePacs. The Semaphore enforces atomic accesses to shared chip-level resources so that the read-modify-write sequence is not broken. The Semaphore block has unique interrupts to each of the CorePacs to identify when that CorePac has acquired the resource.

Semaphore resources within the module are not tied to specific hardware resources. It is a software requirement to allocate semaphore resources to the hardware resource(s) to be arbitrated.

The Semaphore module supports 3 masters and contains 32 semaphores to be used within the system.

There are two methods of accessing a semaphore resource:

- **Direct Access:** A CorePac directly accesses a semaphore resource. If free, the semaphore will be granted. If not, the semaphore is not granted.
- **Indirect Access:** A CorePac indirectly accesses a semaphore resource by writing it. Once it is free, an interrupt notifies the CPU that it is available.

7.30 Antenna Interface Subsystem 2 (AIF2)

The enhanced Antenna Interface subsystem (AIF2) consists of the Antenna Interface module and two SerDes macros. The AIF2 relies on the performance SerDes macro (high-speed serial link) along with a logic layer for the OBSAI RP3 and CPRI protocols. The AIF is used to connect to the backplane for transmission and reception of antenna data, as well as to connect to additional device peripherals.

The AIF2 has 11 timer synchronization events from the AIF2 Timer (AT) module. Timer synchronization events 0-7 are routed as primary events to the EDMA3CC1 and also as secondary events to the C66x CorePacs via CIC0. Timer synchronization events 8, 9, and 10 are hard-wired to TAC, RAC_A, and RAC_B respectively.

Table 7-83 AIF2 Timer Module Timing Requirements (Part 1 of 2)
See Figure 7-52, Figure 7-55, Figure 7-56, and Figure 7-57

No.			Min	Max	Unit
RP1 Clock and Frameburst					
1	tc(RP1CLKN)	Cycle time, RP1CLK(N)	32.55	32.55	ns
1	tc(RP1CLKP)	Cycle time, RP1CLK(P)	32.55	32.55	ns
2	tw(RP1CLKNL)	Pulse duration, RP1CLK(N) low	0.4 * C1 (1)	0.6 * C1	ns
3	tw(RP1CLKNH)	Pulse duration, RP1CLK(N) high	0.4 * C1	0.6 * C1	ns
3	tw(RP1CLKPL)	Pulse duration, RP1CLK(P) low	0.4 * C1	0.6 * C1	ns
2	tw(RP1CLKPH)	Pulse duration, RP1CLK(P) high	0.4 * C1	0.6 * C1	ns
4	tr(RP1CLKN)	Rise time - RP1CLKN 10% to 90%		350.00	ps
4	tf(RP1CLKN)	Fall time - RP1CLKN 90% to 10%		350.00	ps

Table 7-83 AIF2 Timer Module Timing Requirements (Part 2 of 2)
See Figure 7-52, Figure 7-55, Figure 7-56, and Figure 7-57

No.			Min	Max	Unit
4	tr(RP1CLKP)	Rise time - RP1CLKP 10% to 90%		350.00	ps
4	tf(RP1CLKP)	Fall time - RP1CLKP 90% to 10%		350.00	ps
5	tj(RP1CLKN)	Period jitter (peak-to-peak), RP1CLK(N)		600	ps
5	tj(RP1CLKP)	Period jitter (peak-to-peak), RP1CLK(P)		600	ps
6	tw(RP1FBN)	Bit period, RP1FB(N)	8 * C1	8 * C1	ns
6	tw(RP1FBP)	Bit period, RP1FB(P)	8 * C1	8 * C1	ns
7	tr(RP1CLKN)	Rise time - RP1FBN 10% to 90%		350.00	ps
7	tf(RP1CLKN)	Fall time - RP1FBN 90% to 10%		350.00	ps
7	tr(RP1CLKP)	Rise time - RP1FBP 10% to 90%		350.00	ps
7	tf(RP1CLKP)	Fall time - RP1FBP 90% to 10%		350.00	ps
8	tsu(RP1FBN-RP1CLKP)	Setup time - RP1FBN valid before RP1CLKP high	2		ns
8	tsu(RP1FBN-RP1CLKN)	Setup time - RP1FBN valid before RP1CLKN low	2		ns
8	tsu(RP1FBN-RP1CLKP)	Setup time - RP1FBP valid before RP1CLKP high	2		ns
8	tsu(RP1FBN-RP1CLKN)	Setup time - RP1FBP valid before RP1CLKN low	2		ns
9	th(RP1FBN-RP1CLKP)	Hold time - RP1FBN valid after RP1CLKP high	2		ns
9	th(RP1FBN-RP1CLKN)	Hold time - RP1FBN valid after RP1CLKN low	2		ns
9	th(RP1FBN-RP1CLKP)	Hold time - RP1FBP valid after RP1CLKP high	2		ns
9	th(RP1FBN-RP1CLKN)	Hold time - RP1FBP valid after RP1CLKN low	2		ns
PHY Sync and Radio Sync Pulses					
10	tw(PHYSYNCH)	Pulse duration, PHYSYNC high	6.50		ns
11	tc(PHYSYNC)	Cycle time, PHYSYNC pulse to PHYSYNC pulse	10.00		ms
12	tw(RADSYNCH)	Pulse duration, RADSYNC high	6.50		ns
13	tc(RADSYNC)	Cycle time, RADSYNC pulse to RADSYNC pulse	1.00		ms
End of Table 7-83					

1 C1 = tc(RP1CLKN/P)

Figure 7-54 AIF2 RP1 Frame Synchronization Clock Timing

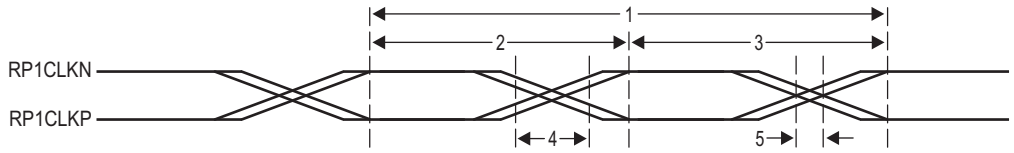


Figure 7-55 AIF2 RP1 Frame Synchronization Burst Timing

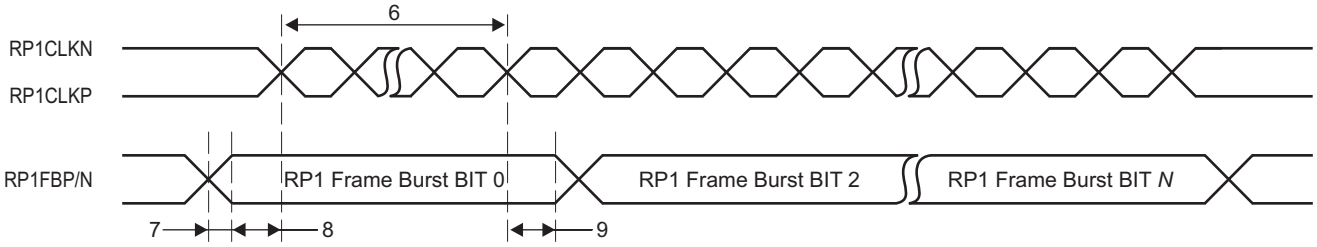


Figure 7-56 AIF2 Physical Layer Synchronization Pulse Timing

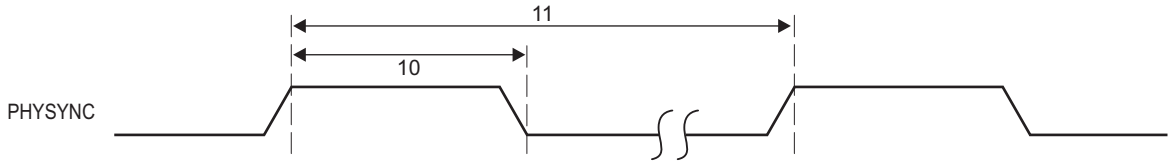


Figure 7-57 AIF2 Radio Synchronization Pulse Timing

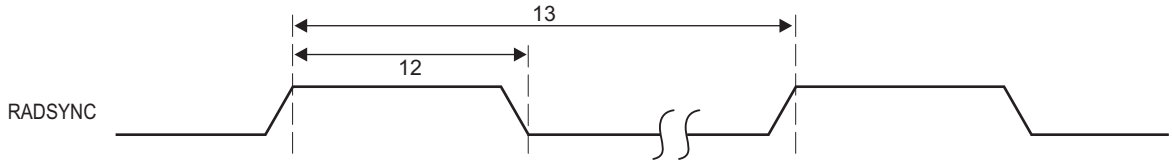
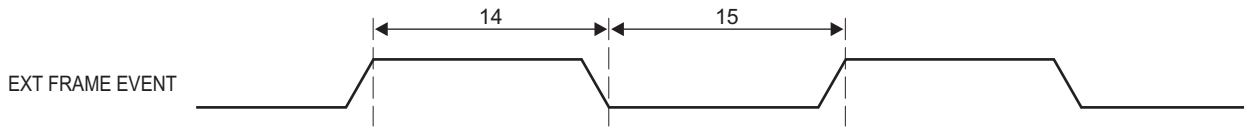


Table 7-84 AIF2 Timer Module Switching Characteristics
(see Figure 7-58)

No.	Parameter		Min	Max	Unit
External Frame Event					
14	tw(EXTFRAMEEVENTH)	Pulse width, EXTFRAMEEVENT output high	4 * C1 (1)		ns
15	tw(EXTFRAMEEVENTL)	Pulse width, EXTFRAMEEVENT output low	4 * C1		ns
End of Table 7-84					

1 C1 = tc(RP1CLKN/P)

Figure 7-58 AIF2 Timer External Frame Event Timing



7.31 Receive Accelerator Coprocessor (RAC)

The TMS320C6670 has two Receive Accelerator Coprocessor (RAC) subsystems. Each RAC subsystem is a receive chip-rate accelerator based on a generic correlator coprocessor (GCCP). It supports UMTS (Universal Mobile Telecommunications System) operations and assists in transferring data received from the antenna to the receive core and performs receive functions that target the W-CDMA macro bits.

The RAC subsystem consists of several components:

- Two GCCP accelerators for finger despread (FD), path monitor (PM), preamble detection (PD), and stream power estimator (SPE).
- Back-end interface (BEI) for management of the RAC configuration and the data output.
- Front-end interface (FEI) for reception of the antenna data for processing and access to all MMRs (memory-mapped registers) and memories in the RAC components.

The RAC has a total of three ports connected to the switch fabric:

- BEI includes two master connections to the switch fabric for output data to device memory. One is 128-bit and the other is 64-bit, both are clocked at CPU/3 rate.
- The FEI has a 64-bit slave connection to the switch fabric for input data as well as direct memory access (to facilitate debug).

7.32 Transmit Accelerator Coprocessor (TAC)

The Transmit Accelerator Coprocessor (TAC) subsystem is a transmit chip-rate accelerator intended to support UMTS (Universal Mobile Telecommunications System) applications.

7.33 Fast Fourier Transform Coprocessor (FFTC)

There are three fast Fourier transform coprocessors (FFTC) intended to accelerate FFT, IFFT, DFT, and IDFT operations. For more information, see the *Fast Fourier Transform Coprocessor (FFTC) for KeyStone Devices User Guide* in 2.9 "Related Documentation from Texas Instruments" on page 66.

7.34 Emulation Features and Capability

7.34.1 Advanced Event Triggering (AET)

The device supports advanced event triggering (AET). This capability can be used to debug complex problems as well as understand performance characteristics of user applications. AET provides the following capabilities:

- **Hardware program breakpoints:** specify addresses or address ranges that can generate events such as halting the processor or triggering the trace capture.
- **Data watchpoints:** specify data variable addresses, address ranges, or data values that can generate events such as halting the processor or triggering the trace capture.
- **Counters:** count the occurrence of an event or cycles for performance monitoring.
- **State sequencing:** allows combinations of hardware program breakpoints and data watchpoints to precisely generate events for complex sequences.

For more information on the AET, see the following documents in 2.9 "Related Documentation from Texas Instruments" on page 66:

- *Using Advanced Event Triggering to Find and Fix Intermittent Real-Time Bugs* application report
- *Using Advanced Event Triggering to Debug Real-Time Problems in High Speed Embedded Microprocessor Systems* application report

7.34.2 Trace

The C6670 device supports trace. Trace is a debug technology that provides a detailed, historical account of application code execution, timing, and data accesses. Trace collects, compresses, and exports debug information for analysis. Trace works in real-time and does not impact the execution of the system.

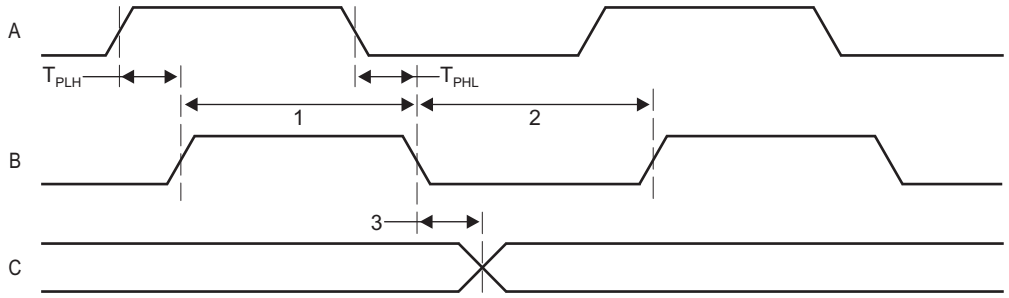
For more information on board design guidelines for trace advanced emulation, see the *Emulation and Trace Headers Technical Reference* in 2.9 "Related Documentation from Texas Instruments" on page 66.

7.34.2.1 Trace Electrical Data/Timing

Table 7-85 Trace Switching Characteristics
(see Figure 7-59)

No.	Parameter		Min	Max	Unit
1	t_w(DPnH)	Pulse duration, DPn/EMUn high	2.4		ns
1	t_w(DPnH)90%	Pulse duration, DPn/EMUn high detected at 90% Voh	1.5		ns
2	t_w(DPnL)	Pulse duration, DPn/EMUn low	2.4		ns
2	t_w(DPnL)10%	Pulse duration, DPn/EMUn low detected at 10% Voh	1.5		ns
3	t_{sko}(DPn)	Output skew time, time delay difference between DPn/EMUn pins configured as trace	-1	1	ns
	t_{skp}(DPn)	Pulse skew, magnitude of difference between high-to-low (tphl) and low-to-high (tplh) propagation delays.		600	ps
	t_{sldp_o}(DPn)	Output slew rate DPn/EMUn	3.3		V/ns
End of Table 7-85					

Figure 7-59 Trace Timing



7.34.3 IEEE 1149.1 JTAG

The JTAG (Joint Test Action Group) interface is used to support boundary scan and emulation of the device. The boundary scan supported allows for an asynchronous TRST (test reset) and only the 5 baseline JTAG signals (e.g., no EMU[1:0]) required for boundary scan. Most interfaces on the device follow the Boundary Scan Test Specification (IEEE1149.1), while all of the SerDes (SRIO and SGMII) support the AC-coupled net test defined in AC-Coupled Net Test Specification (IEEE1149.6).

It is expected that all compliant devices are connected through the same JTAG interface, in daisy-chain fashion, in accordance with the specification. The JTAG interface uses 1.8-V LVCMOS buffers, compliant with the *Power Supply Voltage and Interface Standard for Nonterminated Digital Integrated Circuit Specification* (EAI/JESD8-5).

7.34.3.1 IEEE 1149.1 JTAG Compatibility Statement

For maximum reliability, the C6670 DSP includes an internal pulldown (IPD) on the TRST pin to ensure that TRST will always be asserted upon power up and the DSP's internal emulation logic will always be properly initialized when this pin is not routed out. JTAG controllers from Texas Instruments actively drive TRST high. However, some third-party JTAG controllers may not drive TRST high but expect the use of an external pullup resistor on TRST. When using this type of JTAG controller, assert TRST to initialize the DSP after powerup and externally drive TRST high before attempting any emulation or boundary scan operations.

7.34.3.2 JTAG Electrical Data/Timing

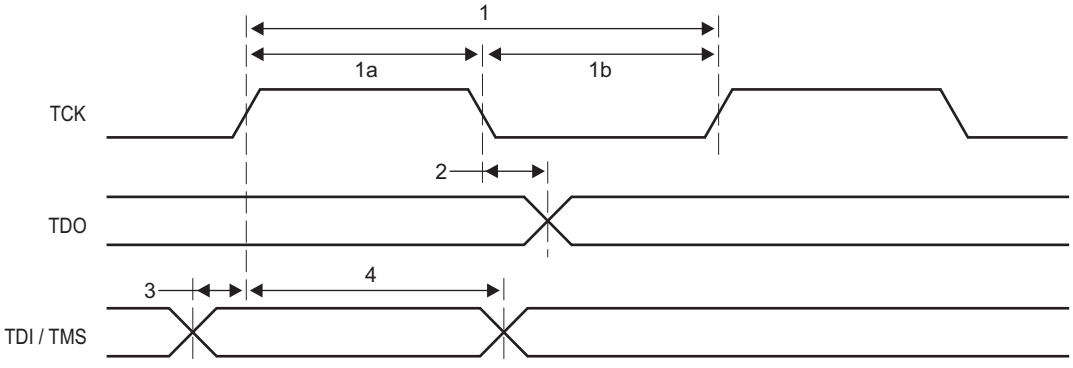
Table 7-86 JTAG Test Port Timing Requirements
(see Figure 7-60)

No.			Min	Max	Unit
1	$t_{c(TCK)}$	Cycle time, TCK	34		ns
1a	tw(TCKH)	Pulse duration, TCK high (40% of tc)	13.6		ns
1b	tw(TCKL)	Pulse duration, TCK low (40% of tc)	13.6		ns
3	tsu(TDI-TCK)	Input setup time, TDI valid to TCK high	3.4		ns
3	tsu(TMS-TCK)	Input setup time, TMS valid to TCK high	3.4		ns
4	th(TCK-TDI)	Input hold time, TDI valid from TCK high	17		ns
4	th(TCK-TMS)	Input hold time, TMS valid from TCK high	17		ns
End of Table 7-86					

Table 7-87 JTAG Test Port Switching Characteristics
(see Figure 7-60)

No.	Parameter		Min	Max	Unit
2	$t_{d(TCKL-TDOV)}$	Delay time, TCK low to TDO valid		13.6	ns
End of Table 7-87					

Figure 7-60 JTAG Test-Port Timing



A Revision History

Revision D

Added PLLSELECT bit to PASSPLLCTL1 Register (Page 146)
Added bridge numbers to the Switch Fabric Connection Matrix tables (Page 87)
Updated DEVSPEED Register (Page 84)
Updated the Interrupt Topology figure (Page 156)
Updated the JTAGID register table (Page 73)
Restricted Output Divide of SECCTL to max value of divide by 2 (Page 132)
Changed TPTCn to EDMA3TCn and TPCCn to EDMA3CCn through-out the document (Page 25)
Marked PBIST_CTL as Reserved (Page 29)
Replaced all CPT with Tracer in the entire document (Page 181)
Replaced all INTC with CIC throughout the document (Page 155)
Updated main PLL lock time (Page 130)
Added PLL Reset bit (Page 146)
Added PLL Reset bit (Page 143)
Marked as Reserved (Page 170)
Marked as Reserved (Page 170)
Added the DDR3 PLL Initialization Sequence (Page 143)
Added the Main PLL and PLL Controller Initialization Sequence (Page 139)
Added the PASS PLL Initialization Sequence (Page 146)
Added po_vcon_smpserr_intr event (Page 162)
Corrected the SPI and DDR3/Hyperbridge Config end addressed (Page 28)

Revision C

Added DEVSPEED Register section (Page 84)
Removed Parameter Information section as the content was not relevant (Page 109)
Added more description to Boot Sequence section (Page 29)
Changed all footnote references from CORECLK to SYSCLK1 (Page 212)
Corrected the typo in the address of MACID2 (Page 207)
Corrected a typo — Changed DDRCLKN to DDRCLKP (Page 144)
Re-arranged the wording for description of SYSCLK1 (Page 110)
Removed example from footnote (Page 178)
Updated footnote on AIF jitter value to 4 ps RMS (Page 141)

Revision B

Changed output skew time for the trace from 500 ps to 1 ns (Page 217)
Corrected description of race condition in DDR3 (Page 194)
Removed all mentions of HHV (Page 110)
Updated description for BWADJ field (Page 138)
Added footnote description for U to UART Timing Requirements (Page 205)
Improved the INTC1 Events Input table (Page 168)
Updated the description for the tc(SPC) parameter (Page 200)
Removed the Max parameters for PHY Sync and Radio Sync Pulses (Page 214)
Added SERDES PLL Status and Config registers (Page 69)
Added table MasterID Settings (Page 179)
Marked event 101 as Reserved (Page 162)
Removed EDMA3 Parameter RAM Memory offset address table. Moved to EDMA UG. (Page 149)
Updated the GMacs and GFlops for 1.2 GHz (Page 13)
Added thermal values into the Thermal Resistance Characteristics table. (Page 221)
Added DDR3PLLCTL1 register and field description table (Page 143)
Added PASSPLLCTL1 register and field descriptions (Page 146)
Added the table of Power Supply to Peripheral I/O Mapping (Page 108)
Marked PREDIV and POSTDIV as reserved registers (Page 131)

Corrected RESET Electrical timing parameters (Page 126)
Removed RESETFULLz parameter from t4b timing description (Page 112)

Revision A

Updated the complete Power-up sequencing section. RESETFULLz must always de-assert after PORz (Page 110)
Updated the description of VARIANT bit field in JTAGID register (Page 73)
Added Setup and Hold times for RP1CLK and RP1CLK signals. (Page 213)
Corrected the size of TETBs for the 4 cores from 16k to 4k (Page 26)
Added RSV0A and RSV0B pins to the Terminal list table (Page 51)
Changed DDR3PLLCTL0 to DDR3PLLCTL and PAPLLCTL0 to PASSPLLCTL (Page 70)
Cleaned up power rail terminology and changed reference parameter in t2c description from t7 to t6 (Page 112)
Added a note on Level Interrupts and EOI values for various modules. (Page 155)
Corrected the address range for I2C MMRs (Page 196)
Corrected Extended Temp max to 100C from 105C (Page 13)
Added BWADJ field to DDR3PLLCTL (Page 143)
Added BWADJ field to PASSPLLCTL (Page 146)

B Mechanical Data

B.1 Thermal Data

Table B-1 shows the thermal resistance characteristics for the PBGA - CYP mechanical package.

Table B-1 Thermal Resistance Characteristics (PBGA Package) [CYP]

No.			°C/W
1	$R\theta_{JC}$	Junction-to-case	0.15
2	$R\theta_{JB}$	Junction-to-board	3.04
End of Table B-1			

B.2 Packaging Information

The following packaging information reflects the most current released data available for the designated device(s). This data is subject to change without notice and without revision of this document.

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead finish/ Ball material (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
TMS320C6670ACYP2	ACTIVE	FCBGA	CYP	841	44	RoHS & Green	SNAGCU	Level-4-245C-72HR	0 to 85	TMS320C6670CYP @2010 TI 1.2GHZ	Samples
TMS320C6670ACYPA	ACTIVE	FCBGA	CYP	841	44	RoHS & Green	SNAGCU	Level-4-245C-72HR	-40 to 100	TMS320C6670CYP @2010 TI A1GHZ	Samples
TMS320C6670ACYPA2	ACTIVE	FCBGA	CYP	841	44	RoHS & Green	SNAGCU	Level-4-245C-72HR	-40 to 100	TMS320C6670CYP @2010 TI A1.2GHZ	Samples
TMS320C6670AXCYP	ACTIVE	FCBGA	CYP	841	44	RoHS & Green	SNAGCU	Level-4-245C-72HR	0 to 85	TMS320C6670XCYP @2010 TI	Samples
TMS320C6670AXCYP2	ACTIVE	FCBGA	CYP	841	44	RoHS & Green	SNAGCU	Level-4-245C-72HR	0 to 85	TMS320C6670XCYP @2010 TI 1.2GHZ	Samples
TMS320C6670AXCYPA	ACTIVE	FCBGA	CYP	841	44	RoHS & Green	SNAGCU	Level-4-245C-72HR	-40 to 100	TMS320C6670XCYP @2010 TI A1GHZ	Samples
TMS320C6670AXCYPA2	ACTIVE	FCBGA	CYP	841	1	RoHS & Green	SNAGCU	Level-4-245C-72HR	-40 to 100	TMS320C6670XCYP @2010 TI A1.2GHZ	Samples
TMS320C6670CYPA2	ACTIVE	FCBGA	CYP	841		RoHS & Green	Call TI	Level-4-245C-72HR	-40 to 100	TMS320C6670CYP @2010 TI A1.2GHZ	Samples
TMS320C6670XCYPA	ACTIVE	FCBGA	CYP	841		RoHS & Green	Call TI	Level-4-245C-72HR	-40 to 100	TMS320C6670XCYP @2010 TI A1GHZ	Samples
TMS320C6670XCYPA2	ACTIVE	FCBGA	CYP	841		RoHS & Green	Call TI	Level-4-245C-72HR	-40 to 100	TMS320C6670XCYP @2010 TI A1.2GHZ	Samples

(1) The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

(2) RoHS: TI defines "RoHS" to mean semiconductor products that are compliant with the current EU RoHS requirements for all 10 RoHS substances, including the requirement that RoHS substance do not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, "RoHS" products are suitable for use in specified lead-free processes. TI may reference these types of products as "Pb-Free".
RoHS Exempt: TI defines "RoHS Exempt" to mean products that contain lead but are compliant with EU RoHS pursuant to a specific EU RoHS exemption.
Green: TI defines "Green" to mean the content of Chlorine (Cl) and Bromine (Br) based flame retardants meet JS709B low halogen requirements of <=1000ppm threshold. Antimony trioxide based flame retardants must also meet the <=1000ppm threshold requirement.

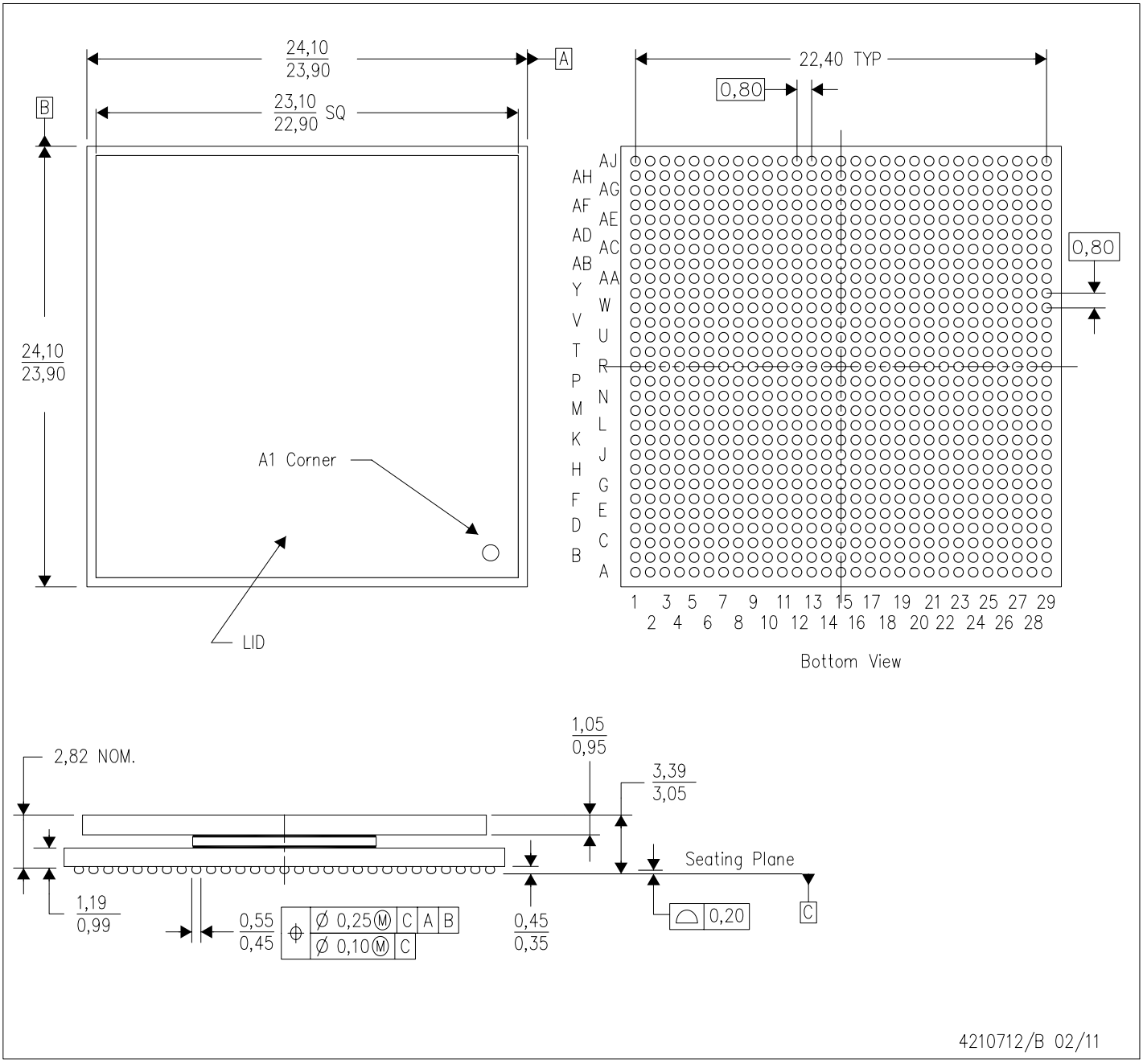
(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

(6) Lead finish/Ball material - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material values may wrap to two lines if the finish value exceeds the maximum column width.

CYP (S–PBGA–N841) PLASTIC BALL GRID ARRAY



NOTES:
A. All linear dimensions are in millimeters. Dimensioning and tolerancing per ASME Y14.5M–1994.
B. This drawing is subject to change without notice.
C. Thermally enhanced plastic package with lid.
D. Flip chip application only.
E. Pb–free die bump and solder ball.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (https:www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265